el BASIC en las aulas

Curso de iniciación

A. Faixes J. Rodá J. Sans

Teoria

94 Ejercicios resueltos

340 Ejercicios propuestos (133 con solución)

PPU

el BASIC en las aulas

Curso de iniciación

A. Faixes J. Rodá J. Sans

PPU
Promociones Publicaciones Universitarias
C/ Nicaragua, 100, 7º1ª. 08029 Barcelona.

Edita

PPU. Promociones Publicaciones Universitarias.
C/ Nicaragua, 100, 7º1ª. 08029 Barcelona.

Ilustraciones

Mª Carmen Berbel

Diseño y composición

Mª Carmen Berbel

Imprime GRAPHIC/2 Lope de Vega 18
08005 Barcelona.

I.S.B.N. 84-86.130-70-0

Dep. Leg. B- 13.018-85

PROLOGO

Este libro pretende ser una iniciación al lenguaje BASIC para principiantes. Desarrollamos dicho lenguaje haciendo referencia a los distintos dialectos del BASIC que se usan para diferentes ordenadores. Estas particularidades deben no obstante consultarse en el manual propio de cada ordenador.

Hemos dividido el libro en diez TEMAS. Cada uno de ellos consta de una EXPLICACION TEORICA acompañada de ejemplos para su mejor comprensión.

Después de cada explicación teórica sigue un apartado con EJERCICIOS RESUELTOS para que el lector vea diferentes técnicas de programación y de aplicación a distintas materias.

Al final de cada tema hay unos EJERCICIOS PROPUESTOS referentes a él. Sólo están las soluciones de los que llevan asterisco excepto los de los dos primeros temas que están íntegras.

El primer tema puede omitirse sin que ello dificulte la comprensión del resto.

Hemos escrito este libro pensando en los siguientes fines:

a) *Que sea LIBRO DE TEXTO para los alumnos de EATP de INFORMATICA de BUP.*

Precisamente este libro es un compendio de las experiencias que tenemos en la docencia de esta asignatura en el INSTITUTO DE BACHILLERATO TORRAS I BAGES de Hospitalet de Llobregat. Nuestros alumnos nos han animado a escribirlo recopilando estas experiencias. A ellos debemos la idea y el ánimo para escribirlo.

En algunos ejercicios del libro no hay que buscar otra utilidad que la didáctica.

b) Que sea también un TEXTO para los alumnos de ESTUDIOS PROFESIONALES que estudian INFORMATICA en las distintas ramas.

Creemos que les puede ser de utilidad tanto las explicaciones teóricas como los ejercicios resueltos y propuestos.

c) A todas las personas que quieran aprender el lenguaje BASIC que es el más utilizado en los ordenadores personales.

No se necesitan conocimientos previos de INFORMATICA para la comprensión de este libro.

En los ejercicios siempre que pidamos un programa o lo exponemos debe tenerse en cuenta que no es único. Todo programa admite variantes que dependen no sólo de la capacidad del que lo escribe, sino también de la forma de presentación que se desee.

Programar es una actividad creativa, la imaginación y el ingenio del programador juegan en ello un papel importante.

Todo programa admite mejoras. Es frecuente que al abordar un mismo problema en días diferentes, el propio programador efectúe variantes en su programación.

Unas veces porque se da cuenta de que pueden evitarse algunos pasos y así tener más memoria disponible en su ordenador y otras veces porque la presentación del mismo no le parece suficientemente clara o bien puede mejorarse.

Un programa puede darse por acabado cuando ha pasado por el proceso de depuración de errores y perfeccionamiento.

La mayoría de los programas expuestos en el libro admiten este proceso de perfeccionamiento, que en algunos casos hemos evitado para mejor comprensión.

Téngase en cuenta por otra parte, que a medida que se aprenden nuevas sentencias se pueden modificar los programas, resultando generalmente más rápidos o más vistosos o ambas cosas a la vez, como ocurre con problemas repetidos en algunos capítulos.

Nuestra experiencia en la docencia de las Matemáticas nos ha podido llevar a poner ejemplos y ejercicios sobre temas de Matemáticas. No obstante nos hemos esforzado en corregir este punto insertando ejercicios referentes no sólo a Matemáticas sino a otras materias y también a aplicaciones no escolares.

Agradecemos a nuestros compañeros de docencia todas las observaciones y sugerencias que han tenido a bien hacernos durante la elaboración de este libro, asímismo agradecemos la labor realizada por la Sra. Mª Carmen Berbel en la composición del libro y en la realización de los dibujos.

No hacemos referencia en el libro a las representaciones gráficas puesto que las posibilidades gráficas y métodos de uso dependen de cada modelo.

Barcelona, Febrero de 1985.

LOS AUTORES,

1
BREVE HISTORIA DEL ORDENADOR. SISTEMAS DE NUMERACION

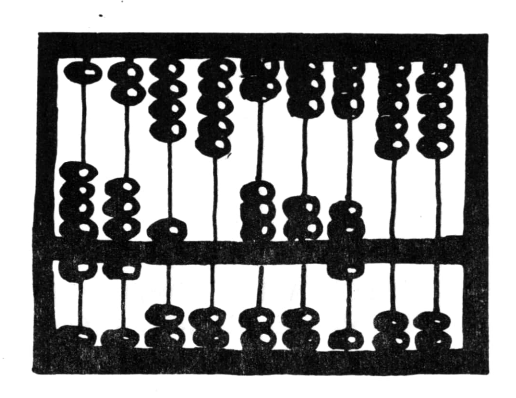
1. Breve historia del ordenador

Una de las necesidades que ha tenido el hombre ha sido contar objetos y atribuir guarismos al resultado de esta operación.

No queremos hacer ciencia ficción de cómo nace el concepto de número y de cómo evoluciona a través de la prehistoria, aunque hay caminos para hacer un estudio científico de lo que pudo pasar, por ejemplo observar entidades étnicas cuya evolución está aún muy atrasada y a partir de este estudio rellenar los eslabones que nos separan de estos grupos étnicos.

Supondremos pues que el hombre ya conoce el concepto de número abstracto y que además realiza operaciones con estos números. Escoge una base para contar y podemos pensar que ya está utilizando la base diez, lo que le ayuda a contar con los dedos de las manos y así utilizar la primera calculadora natural.

En el momento de hacer operaciones, el hombre busca la manera de realizarlas con más rapidez. Lo consigue con el **ABACO**, cuyo fundamento es un conjunto de unas piezas ensartadas en varillas, y cuyo uso ha llegado hasta nuestros días. Con esta máquina se pueden realizar con facilidad operaciones de suma y resta.



Abaco

John NEPER (nacido en 1550) da un paso más en conseguir rapidez en los cálculos con el descubrimiento de los logaritmos.

Henry BRIGGS (nacido en 1556) escribe las primeras tablas de logaritmos para el uso práctico de los mismos.

Si bien es cierto que el uso de los logaritmos da mayor rapidez en el momento de efectuar las operaciones, sin embargo, no constituyen una "máquina de calcular" propiamente dicha.

En la misma época, **William OUGHTRED** (nacido en 1574) utiliza dos escalas de logaritmos que superponiéndolas y deslizándolas una sobre otra efectúa cálculos mecánicos con ayuda de logaritmos. Esta pequeña "máquina de calcular" se conoce como Regla de Cálculo y su uso ha sido enormemente difundido hasta llegar las máquinas de calcular electrónicas.

Más tarde, **Blaise PASCAL** (nacido en 1623), diseña la primera máquina de calcular mecánica cuya estructura se basa en ruedas dentadas unidas entre sí de forma que 10 dientes de la 1ª hacen correr un diente de la 2ª y así sucesivamente. Con esta máquina se podía sumar y restar.

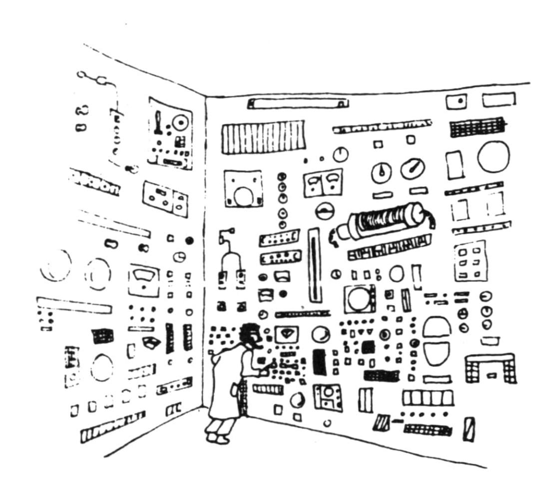
Gottfried LEIBNIZ (nacido en 1646) mejoró la máquina de PASCAL construyendo otra que además de sumar y restar multiplicaba y dividía.

Charles BABAGE (nacido en 1792) ideó una máquina que podía funcionar por medio de fichas perforadas y sus proyectos son básicos para las calculadoras modernas, sin embargo no pudo realizar sus proyectos debido a su complejidad técnica.

A partir de este momento se perfeccionan cada día más las máquinas de calcular mecánicas, pero adolecen de la complejidad de los mecanismos que tienen que usar.

Hacia 1940 se construyen las primeras máquinas de calcular electrónicas. Efectuan las operaciones en el sistema binario, que sólo tiene dos símbolos 0 y 1 que en la máquina es: no pasa corriente (0) o pasa corriente (1).

En 1944 se construyó el Mark 1 que era una máquina enorme con relés y tubos de vacío. Rápidamente se construye el ENIAC que es el principio de la **1ª generación de ordenadores.** Funcionaban con válvulas electrónicas y tenían muchos problemas de calentamiento haciendo costoso el mantenimiento.



Hacia 1957 las válvulas electrónicas son sustituídas por transistores lo que supone un gran salto hacia la miniturización y velocidad de los ordenadores. Se dice que estos ordenadores forman la **2ª generación.**

En la década de los 60 aparecen los ordenadores de la **3ª generación** que sustituyen los transistores por circuitos integrados que tienen un tamaño similar a los transistores pero pueden contener centenares de componentes elementales.

Actualmente se considera que nos encontramos en la **4ª generación** de ordenadores que utilizan circuitos integrados con millares de componentes elementales en su interior.

2. Sistema de numeración. Sistema binario

En base diez utilizamos diez símbolos: 0,1,2,3,...,9 (guarismos) que significan cantidades distintas según la posición, por ejemplo:

$$2521 \quad \text{significa} \quad 1+2.10+5.10^2+2.10^3$$

En general cuando escribimos h ... cba significa $a+b.10+c.10^2+...+h.10^i$ (i es el número de guarismos que hay en h...cba disminuído en una unidad)

Si sólo utilizamos cuatro guarismos: 0,1,2,3, diremos que trabajamos en base cuatro.

Por ejemplo el número

$$213_{(4}$$

significa $3+1.4+2.4^2$ escrito en base diez.

En general:

$$h...cba_{(n} = a + b.n + c.n^2 +...+ h.n^i$$

escrito en base diez.

a) Paso de un número en base n a base decimal

Dado un número $N_{(n}$, para saber cual es su expresión en el sistema decimal, nos basta efectuar las operaciones que expresan dicho número en su forma polinómica.

Ejemplo

$$51462_{(7} = 2 + 6.7 + 4.7^2 + 1.7^3 + 5.7^4 = 12588$$

b) Paso de un número en base 10 a base n

Para ello basta efectuar divisiones sucesivas por dicha base y quedarnos con los restos parciales hasta que el último cociente sea menor que n.

Es decir, si queremos escribir en número 7456 en base 8, procederemos de la siguiente manera:

```
7456  |8
  25  |932   |8
   16  13    |116   |8
    0    52    36   |14   |8
          4     4     6   |1
 ↓        ↓     ↓     ↓    ↓
1ª cifra  2ª cifra  3ª cifra  4ª cifra  5ª cifra
derecha   derecha   derecha   derecha   derecha
```

o sea:

$$7456 = 16440_{(8}$$

En caso de utilizar una base n mayor que diez, entonces nos auxiliamos de letras. Por ejemplo en la base 12 los guarismos empleados para los números 10 y 11 son las letras α y β.

c) Paso de un número en base n a otra base m

Dicho paso se efectúa a través de la base diez, es

decir, pasamos primero el número en base n a base diez y luego por sucesivas divisiones a base m.

Por ejemplo, para pasar $12034_{(5}$ a base 9 es:

$$12034_{(5} = 4 + 3.5 + 0.5^2 + 2.5^3 + 1.5^4 = 894$$

```
894 |9
 84 |99  |9
  3  09  |11  |9
      0    2  |1
```

Por lo que

$$12034_{(5} = 1203_{(9}$$

Para pasar de base 8 (Sistema octal) a base 2 (Sistema binario) en la práctica es cómodo hacerlo de la siguiente manera:

Se pasa cada cifra del sistema octal a base 2 escribiendo el resultado con 3 cifras (para completar poner ceros a la izquierda). El número resultante de juxtaponerlos es el número en base 2.

<u>Ejemplo</u>

$$\underbrace{1}_{001} \quad \underbrace{4}_{100} \quad \underbrace{2}_{010} \quad \underbrace{1}_{001}$$

De donde:

$$1421_{(8} = 1100010001_{(2}$$

Siguiendo el proceso inverso podríamos pasar del sistema binario al octal.

Análogamente se haría para pasar del sistema de base 16 (Sistema hexadecimal) al sistema binario, en este caso cada cifra del sistema hexadecimal hay que escribirlo con cuatro cifras del binario añadiendo

a la izquierda los ceros necesarios (ver el ejemplo I.11.b)

Ejemplo

¿En qué sistema de numeración de base x, $x \in \mathbb{N}$, el número 101 en base diez tiene como expresión $10202_{(x}$

En este caso debe verificarse:

$$2 + 0.x + 2.x^2 + 0.x^3 + 1.x^4 = 101 ,$$

o sea:

$$x^4 + 2x^2 - 99 = 0$$

cuyas soluciones son ± 3

Se trata pues de la base 3

d) Operaciones en una base cualquiera

Se efectúan siguiendo los mismos métodos que en el sistema decimal.

Ejemplo

Efectuar la suma

$$\begin{array}{r} 2305_{(6} \\ 1054_{(6} \\ +\ 4432_{(6} \\ \hline \end{array}$$

La solución es $12235_{(6}$

Ejemplo

Efectuar el producto

$$\begin{array}{r} 46315_{(7} \\ \times\ 3_{(7} \\ \hline \end{array}$$

La solución es $205251_{(7}$

e) Sistema binario

Se llama así al que tiene como base $n=2$.

En este caso todo número consta solamente de los guarismos 0 y 1.

<u>Ejemplo</u>

El número 7521 en el sistema binario es:

```
7521   |2
15     |3760   |2
 12     17     |1880   |2
  01    16      08     |940   |2
   1     00      00     14    |470   |2
          0       0      00    07    |235   |2
                          0    10     03   |117   ...
                                0      15
                                        1

...   117  |2
       17  |58   |2
        1   18   |29   |2
             0    09   |14   |2
                   1     0   |7   |2
                              1   |3   |2
                                   1   |1
```

Es decir:

$$7521 = 1110101100001_{(2}$$

Si bien la expresión de los números en el sistema binario es larga, tiene la ventaja de usar sólo dos cifras que pueden simularse por ejemplo en un circuito con un 1 si pasa la corriente y por un 0 si no pasa la corriente. Este es precisamente el sistema que usa el ordenador para efectuar los cálculos, aunque para exponerlos en pantalla nos lo traduzca al sistema decimal.

EJERCICIOS RESUELTOS

I.1. *Escribir el número decimal 124 en base 4.*

Solución

Efectuando las sucesivas divisiones:

$$\begin{array}{r|l} 124 & 4 \\ \hline 04 & 31 \\ 0 & 3 \end{array} \quad \begin{array}{r|l} 31 & 4 \\ \hline 3 & 7 \end{array} \quad \begin{array}{r|l} 7 & 4 \\ \hline 3 & 1 \end{array}$$

De donde:

$$124 = 1330_{(4}$$

* * *

I.2. *Escribir $230_{(4}$ en base diez*

Solución

Se tiene:

$$230_{(4} = 3.4 + 2.4^2 = 44$$

* * *

I.3. *Escribir en base 2 el número $425_{(6}$*

Solución

Pasando a base diez:

$$425_{(6} = 5. + 2.6 + 4.6^2 = 161$$

Dicho número en base 2 es:

```
161  |2
 01  |80  |2
  1    0  |40  |2
             0 |20  |2
                  0 |10  |2
                        0 |5  |2
                            1 |2  |2
                                0 |1
```

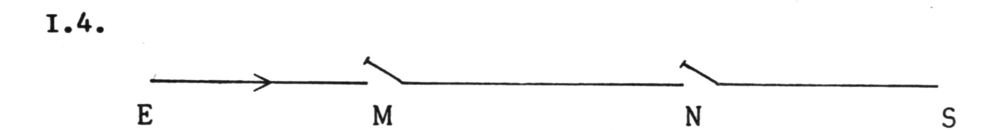
$$161 = 10100001_{(2}$$

O sea:

$$425_{(6} = 10100001_{(2}$$

* * *

I.4.



El gráfico adjunto representa un conductor de corriente en el cual hay dos interruptores (M y N); cuando por la salida (S) pasa corriente se escribe 1 y cuando no pasa el 0. Si A significa "interruptor abierto" y C "interruptor cerrado" escribir en base 10 el número escrito en base 2 que resulta del siguiente esquema:

M N	M N	M N	M N	M N	M N
C C	A C	A A	C C	A A	C A

Solución

El número escrito en base 2 es:

$$100100_{(2}$$

En base diez:

$$100100_{(2} = 0 + 0.2 + 1.2^2 + 0.2^3 + 0.2^4 + 1.2^5 = 4+32 = 36$$

* * *

I.5. *Efectuar la suma de los siguientes números escritos en base 8:*

$$246_{(8}\ ,\ 127_{(8}$$

Solución

$$\begin{array}{r} 246_{(8} \\ +\ 127_{(8} \\ \hline 375_{(8} \end{array}$$

* * *

I.6. *Efectuar la resta* $1110_{(2} - 1001_{(2}$

Solución

$$\begin{array}{r} 1110_{(2} \\ -\ 1001_{(2} \\ \hline 101_{(2} \end{array}$$

* * *

I.7. *Calcular el producto $345_{(6} \times 235_{(6}$*

Solución

$$
\begin{array}{r}
345 \\
\times\ 235 \\
\hline
3101 \\
1523 \\
1134 \\
\hline
140131
\end{array}
$$

O sea:

$$345_{(6} \times 235_{(6} = 140131_{(6}$$

* * *

I.8. *Efectuar el cociente 31405: 232 en base 6.*

Solución

Teniendo en cuenta la siguiente tabla (en base 6):

$$
\begin{aligned}
232 \times 1 &= 232 \\
232 \times 2 &= 504 \\
232 \times 3 &= 1140 \\
232 \times 4 &= 1412 \\
232 \times 5 &= 2044
\end{aligned}
$$

se tiene:

$$
\begin{array}{l|l}
31405 & 232 \\
\cline{2-2}
\underline{232} & 114 \\
420 & \\
\underline{232} & \\
1445 & \\
\underline{1412} & \\
033 &
\end{array}
$$

Por tanto:

Cociente: $114_{(6}$

Resto: $33_{(6}$

* * *

I.9. *¿En qué base se verifica la igualdad: 63+56=141?*

Solución

Sea x esta base. Para que se cumpla la igualdad debe ser:

$$6x + 3 + 5x + 6 = x^2 + 4x + 1$$

$$x^2 - 7x - 8 = 0$$

$$x = \frac{7 \pm \sqrt{49+32}}{2} = \frac{7 \pm \sqrt{81}}{2} = \frac{7 \pm 9}{2} = \begin{cases} \nearrow 8 \\ \searrow -1 \end{cases}$$

Como debe ser $x > 0$ la base es $x=8$.

* * *

I.10. *Expresar en el número $0,\widehat{6}$ en base 6 sabiendo que está escrito en el sistema decimal.*

Solución

Se tiene que:

$$0,\widehat{6} = 6/9 = 2/3$$

Efectuando la división en base 6 y teniendo en cuenta la

tabla:

$$3 \times 1 = 3$$
$$3 \times 2 = 10$$
$$3 \times 3 = 13$$
$$3 \times 4 = 20$$

da:

$$\begin{array}{r|l} 2{,}0 & 3 \\ \hline 0 & 0{,}4 \end{array}$$

es decir:

$$0{,}\widehat{6} = 0{,}4_{(6}$$

* * *

I.11. *Expresar en el sistema binario los números*

a) $145_{(8}$ **b)** $198_{(16}$

<u>Solución</u>

a)

$$\underbrace{001}_{1} \quad \underbrace{100}_{4} \quad \underbrace{101}_{5}$$

De donde:

$$145_{(8} = 1100101_{(2}$$

b)

$$\underbrace{0001}_{1} \quad \underbrace{1001}_{9} \quad \underbrace{1000}_{8}$$

De donde:

$$198_{(16} = 110011000_{(2}$$

* * *

I.12. *Expresar 7458 en base 2 pasando previamente a base 8.*

<u>Solución</u>

$$
\begin{array}{rl}
7458 & \underline{|\,8\quad} \\
25 & |\,932 \quad \underline{|\,8\quad} \\
18 & \;\;13 \quad\;\; |\,116 \quad \underline{|\,8\quad} \\
2 & \;\;\;\;52 \qquad 36 \quad |\,14 \quad \underline{|\,8\quad} \\
 & \;\;\;\;\;\;4 \qquad\;\; 4 \qquad\; 6 \quad |\,1
\end{array}
$$

Por tanto:

$$7458 = 16442_{(8}$$

$$\overbrace{001}^{1} \qquad \overbrace{110}^{6} \qquad \overbrace{100}^{4} \qquad \overbrace{100}^{4} \qquad \overbrace{010}^{2}$$

Luego

$$7458 = 1110100100010_{(2}$$

* * *

EJERCICIOS PROPUESTOS

1. En el sistema de base 6, ¿existe el guarismo 6?

2. Contestar a las siguentes preguntas dando las respuestas en base 2:
a) ¿Cuántas manos tenemos?
b) ¿Cuántos dedos tenemos en una mano?
c) ¿Cuántos días tiene una semana?
d) ¿Cuántos meses tiene el año?

3. Escribir en base diez el número $10011101_{(2}$

4. Escribir en base 3 el número $10010_{(2}$

5. Escribir 324 en base 2.

6. ¿Cuál es la expresión del número 75418 en el sistema de base: a) 8; b) 2 ?

7. Dados los números 346 y 250 en base 7, escribirlos en base 2.

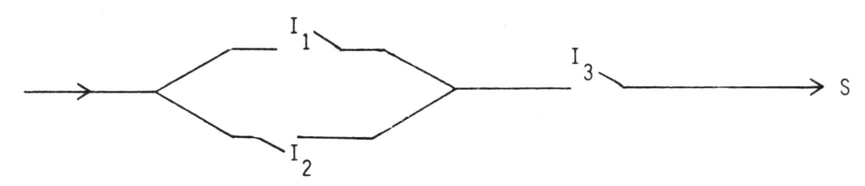
8. Dados los números 475 y 274 en base 8, escribirlos en base 4.

9. ¿Cuál es la expresión del número 496158 en base 12?

10. ¿Cuál es la expresión del número 7585 en el sistema de base: a) 6; b) 12 ?

11. Si $\alpha=10$, $\beta=11$, $\gamma=12$, escribir en base diez el número $\alpha\alpha\beta\beta\gamma\gamma_{(13}$

12.



El gráfico adjunto representa un conductor de corriente en el cual hay los interruptores I_1, I_2, I_3. Cuando por la salida (S) pasa corriente se escribe 1 y cuando no pasa el 0. Si A significa "interruptor abierto" y C "interruptor cerrado", escribir en base 10 el número escrito en base 2 que resulta del siguiente esquema:

$I_1I_2I_3$	$I_1I_2I_3$	$I_1I_2I_3$	$I_1I_2I_3$	$I_1I_2I_3$
A C C	C C A	C A A	A A C	C A C

13. Escribir las tablas de sumar y de multiplicar en el sistema de base 7.

14. Efectuar la siguiente suma: $1001_{(2} + 1101_{(2}$

15. Efectuar la siguiente suma: $3214_{(5} + 2014_{(5} + 3323_{(5}$ Dar el resultado en base 2.

16. Calcular $1011001_{(2} - 11101_{(2}$

17. Calcular $72142_{(8} - 64354_{(8}$

18. Efectuar el producto $1011101_{(2} \times 110011_{(2}$. ¿Cuál es el resultado en el sistema decimal?

19. Efectuar el cociente $41742_{(8} : 315_{(8}$

20. ¿En qué base se verifica la igualdad: $122 \times 21 = 3222$?

21. ¿En qué base se verifica la igualdad: $143 \times 32 = 5016$?

22. ¿En qué base se verifica la igualdad: $512-404 = 104$?

23. En una cierta base se verifica que $244+567 = 822$. Hallar en dicha base 371×456.

24. Resolver la ecuación $7x+18 = 122-5x$ en base 9.

25. Resolver la ecuación $4x+52 = 126-7x$ en base 8.

26. Hallar x sabiendo que $546 = 20202_{(x}$

27. Hallar x sabiendo que $90 = 10100_{(x}$

28. Expresar el número $0,\widehat{39}$ en base 6 (con tres decimales).

2
EL ORDENADOR
SU USO COMO CALCULADORA

1. Partes de un ordenador

El ordenador está formado por las siguientes partes:

- **La Unidad Central de Proceso** (C.P.U.)
- **Memoria**

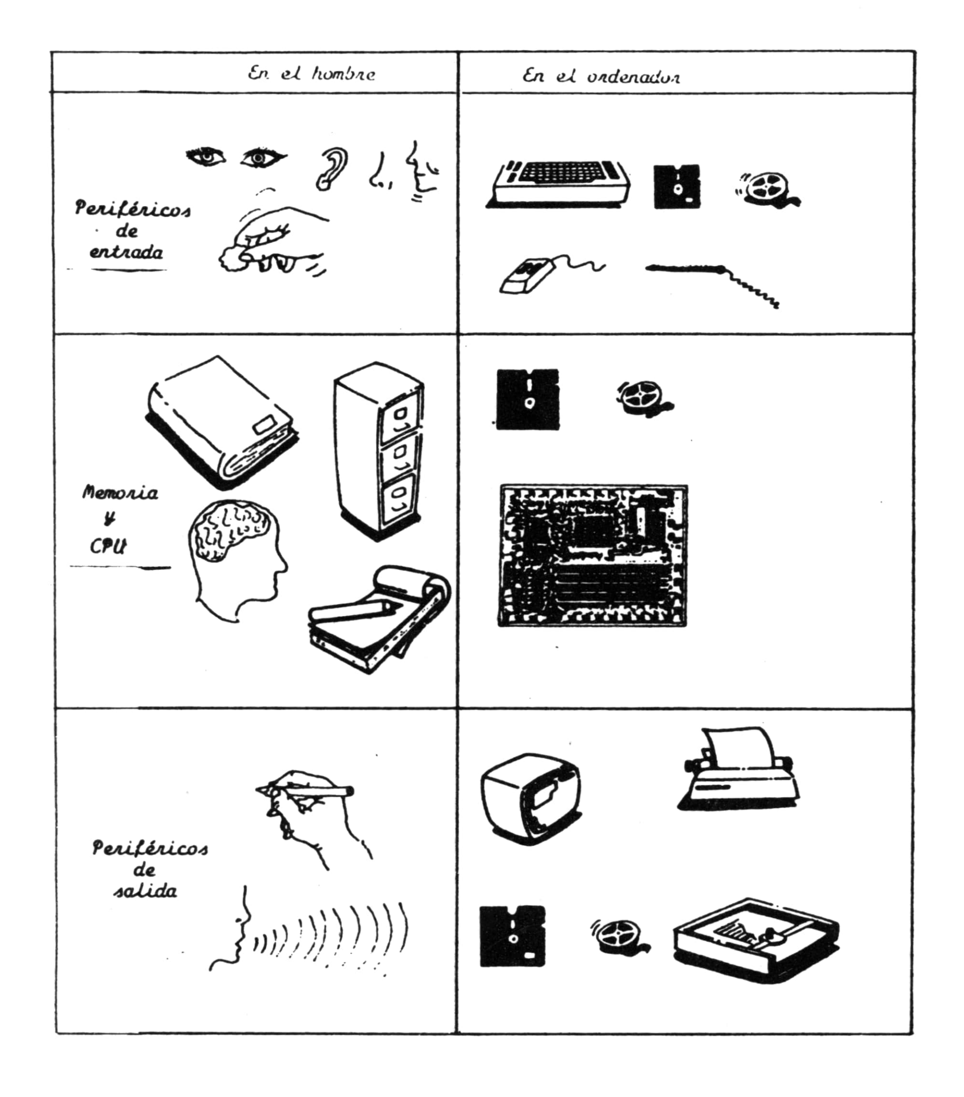
La C.P.U. consta de:

- **Unidad de Control** (U.C.), cuya función es distribuir y controlar el trabajo que han de realizar otras partes del ordenador.
- **Unidad Lógica Aritmética** (U.L.A.), cuya función es realizar operaciones aritméticas y comparaciones entre los datos.

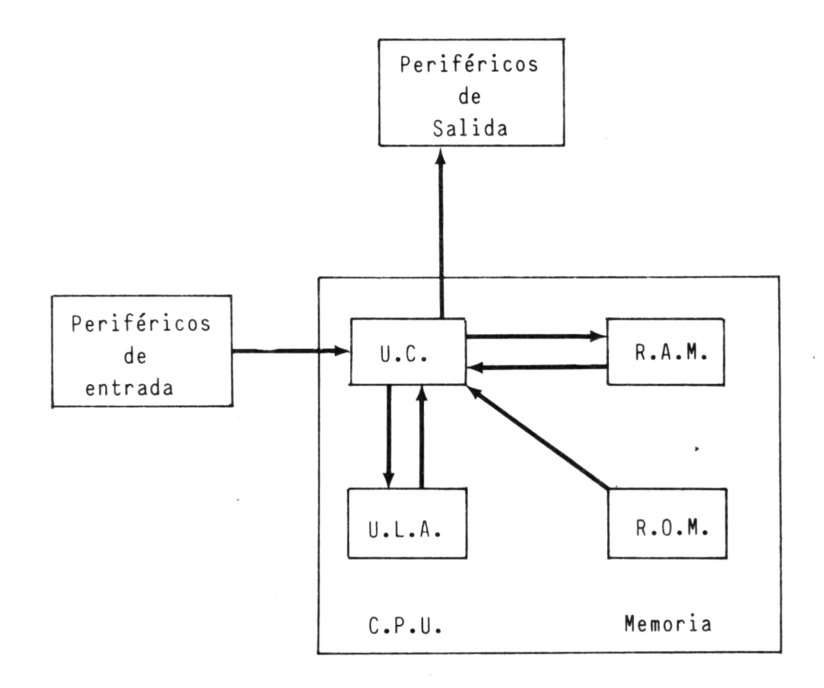
La Memoria está formada por:

- **Memoria R.A.M.** (Random Acces Memory). Es la que almacena la información dada por el usuario (datos, programas,...). Guarda los programas y resultados intermedios. El usuario puede modificar el contenido de la memoria RAM. En el momento de desconectar el ordenador queda borrado todo lo que contenía.
- **Memoria R.O.M.** (Read Only Memory). Es una memoria fija o sea no alterable por el usuario. En ella el fabricante incluye funciones y programas para ser utilizados. En el momento de desconectar el ordenador no se altera el contenido de la memoria ROM.

El usuario se comunica con el ordenador a través de los periféricos, que pueden ser: **Periféricos de Entrada** (teclado, cintas, discos,...) y **Periféricos de Salida** (pantalla, impresora, cintas, discos,...).


En el hombre
En el ordenador
Periféricos de entrada
Memoria y CPU
Periféricos de salida

Un esquema general es:

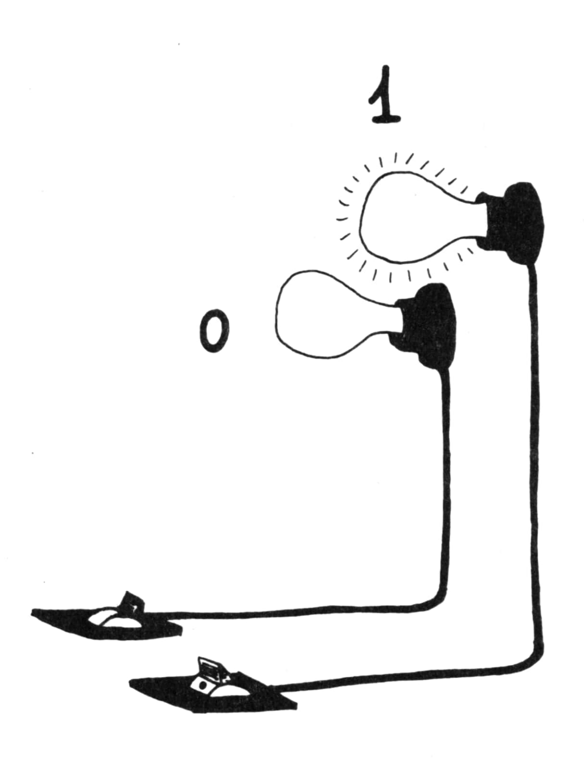


ORDENADOR

A los periféricos que sirven para almacenar programas y datos que pueden volver a introducirse en el ordenador se les llama **Memorias de Masa.**

2. Definiciones

a) **BIT**
Se llama así a la unidad más pequeña en el lenguaje del ordenador y sólo puede tomar dos valores, 0 y 1.



Bit

0 *Si no pasa la corriente.*

1 *Si pasa la corriente.*

b) **BYTE**
Se llama así al grupo de 8 bits (octeto)

c) **KILO-BYTE**
Lo forman 1024 = 2^{10} bytes, o sea 8192 bits. La capacidad de memoria de un ordenador viene dada generalmente por el número de K-bytes del mismo.

d) **HARDWARE**
Se designa con este nombre al ordenador como máquina. Lo constituyen los componentes físicos del ordenador.

e) **SOFTWARE**
Se designa con este nombre a los elementos que forman parte de la programación: lenguajes, diagramas, métodos de programación, etc., que se estudiarán en el tema III.

3. El ordenador como calculadora

Para utilizar el ordenador como calculadora (en el lenguaje BASIC), basta escribir el mensaje PRINT, ya sea con una tecla especial ya sea deletreando la palabra, según el modelo de ordenador que se use.

Así pues,

PRINT 5+4

nos dará en la pantalla 9

PRINT 7*6

nos dará en la pantalla 42

Decimos que la pantalla nos dará el resultado, pero esto significa que deberá pulsarse la tecla que pone en funcionamiento la orden dada al ordenador. Esta tecla depende del modelo usado y puede ser: RETURN, EXE, ENTER, NEW LINE, etc.

4. Operadores aritméticos

Los símbolos que se usan para las operaciones aritméticas son:

+ Para la suma
- Para la diferencia
* Para el producto
/ Para el cociente
↑ Para la potencia
(también se usa ** y ∧)

siendo la prioridad de operaciones la siguiente:

1º La potencia ↑

2º El producto y el cociente (* y /) en el orden en que estén escritas.

3º La suma y la diferencia (+ y -) en el orden en que estén escritas.

No olvidemos el uso de los paréntesis para operaciones combinadas

Por ejemplo:

Para realizar $\left(4 + \frac{3^3 \cdot 16}{64} + 8\right) \cdot 51$ pulsaremos

```
PRINT (4 + 3 ↑ 3 * 16/64 + 8)*51
```

El resultado en la pantalla es (después de pulsar RETURN ó ENTER, etc...):

956.25

No nos debe extrañar ver un punto en vez de una coma, pues el ordenador separa los decimales por un punto, ignorando este significado para la coma. Así mismo, el cero se representa por Ø para diferenciarlo de la letra O. Es fácil al principio cometer estos dos fallos con lo cual el ordenador nos dará un mensaje de error en el caso de equivocarnos.

Las potencias deben ser de base positiva pues de lo contrario también nos daría mensaje de error. Algunos

ordenadores calculan potencias de exponente entero y base negativa. Nosotros en los ejercicios supondremos que no lo hacen.

Para operar con números muy grandes o muy pequeños se utiliza la **notación científica** (potencias de 10)

Ejemplos

1) 2.316 E+24 significa $2,316 \cdot 10^{24}$

2) 1.7432 E-9 significa $1,7432 \cdot 10^{-9}$

También se pueden entrar datos con dicha notación.

El intervalo numérico de cálculo del ordenador va desde 10^{-38} hasta 10^{38}.

5. Funciones Matemáticas

Las funciones que normalmente nos encontramos en el ordenador son:

SGN: signo
- = +1 si el argumento es positivo
- = -1 si el argumento es negativo
- = 0 si el argumento es cero

NOTA: Cuando hablamos del "argumento" de una función nos referimos a la expresión a la que se aplica la función. Por ejemplo:

PRINT	SGN	(7*5 - 4*1∅)
(sentencia)	(función)	(argumento)

ABS: valor absoluto (Halla el valor absoluto del argumento)
INT: parte entera (atención : INT(-2.3) = -3)
SQR: raíz cuadrada (podríamos escribir también ↑∅.5)
EXP: exponencial de base e
LOG ó LN: Logaritmo neperiano

SIN: seno (dando el argumento en radianes)
COS: coseno (dando el argumento en radianes)
TAN: tangente (dando el argumento en radianes)
ASN: arco seno (el resultado en radianes)
ACS: arco coseno (el resultado en radianes)
ATN: arco tangente (el resultado en radianes)

En algunos ordenadores no están definidas las funciones arc sen y arc cos.

Si en una expresión hay funciones y operaciones aritméticas, la prioridad es:

1º Funciones
2º Operaciones aritméticas

Estas funciones se utilizan con el siguiente formato

```
PRINT ...    (...)
  ↑            ↑
función    argumento
```

Ejemplos

1) PRINT EXP(4) da como resultado 54.59815 que es e^4

2) PRINT EXP 4*2 da como resultado 1Ø9.1963 que es $e^4.2$

3) PRINT EXP(4*2) da como resultado 298Ø.958 que es e^8

4) PRINT SIN(π)/2 da como resultado Ø que es

$$\frac{\text{sen } \pi}{2} = \frac{\emptyset}{2} = \emptyset$$

Algunos ordenadores no tienen una tecla para el número π (Pi). En este caso debe sustituirse π(Pi) por una aproximación, por ejemplo 3.1415927. (Generalmente admiten la aproximación de π con menos decimales).

5) PRINT SIN(π/2) da como resultado 1 que es

$$sen(\frac{\pi}{2}) = 1$$

Puesto que las funciones trigonométricas operan con radianes hay que tener en cuenta que:

$$\begin{cases} \text{n grados equivalen a } (n*\pi/180) \text{ radianes} \\ \text{n radianes equivalen a } (n*180/\pi) \text{ grados} \end{cases}$$

Por lo que si queremos hallar sen 45º, pulsaremos:

PRINT SIN (45*π/18Ø)

la respuesta en pantalla es:

Ø.7Ø71Ø678

Por otra parte, si queremos buscar el ángulo correspondiente a una cierta función trigonométrica, por ejemplo, para hallar $x = \text{arc tag} \sqrt{3}/3$ en grados, pulsaremos:

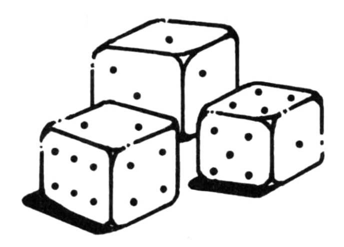
PRINT ATN (SQR (3)/3)*18Ø/π

la respuesta en la pantalla es: 3Ø

Evidentemente que si nos interesa el resultado en radianes, sobra el factor 18Ø/π.

6. Simulación de números aleatorios

En algunos casos interesa que el ordenador nos de una serie de número aleatorios, es decir, dados al azar como pudiera ser los números de un sorteo o de un juego de dados.



Esto se consigue con la función RND. Esta función da un número x al azar comprendido entre Ø y 1 $(0 \leqslant x < 1)$

Si nos interesa tener números aleatorios mayores nos bastará combinar esta función con el producto. Si queremos por ejemplo un número aleatorio n $(Ø \leqslant n < 1Ø)$ bastará pulsar

```
PRINT 1Ø * RND
```

RND es una función sin argumento.

En ocasiones no nos interesa la parte decimal de este número aleatorio; pensamos por ejemplo en la simulación de un juego de dados donde sólo es posible que salgan los números naturales del 1 al 6. En este caso nos bastará pulsar

```
PRINT 1 + INT (6 * RND)
```

que nos dará el resultado pedido, puesto que 6*RND nos dará un número al azar entre Ø y 6 $(Ø \leqslant n < 6)$. Al tomar su parte entera nos quedaremos con un número al azar entre Ø y 5 (ambos incluidos), por lo que al añadirle 1 nos dará un número n:

$$1 \leqslant n \leqslant 6$$

Los números generados por la función RND siguen en realidad una cierta secuencia, por esto se les llama **pseudoaleatorios**.

EJERCICIOS RESUELTOS

II.1. *Calcular*

$$\frac{42176810 + 234}{2^{10} - 10247}$$

Solución

Debe pulsarse:

```
PRINT (42176810+234)/(2↑10-10247)
```

que imprime:

```
-4573.0287
```

* * *

II.2. *Calcular*

a) $\dfrac{2,3.10^{5} + 4,71.10^{-2}}{42.10^{-3}}$ b) $\dfrac{9,8.10^{14} + 5,876.10^{5}}{9,987.10^{5} - 10^{6}}$

Solución

a) Debe pulsarse:

```
PRINT (2.3 * 10 ↑ 5 + 4.71 * 10 ↑(-2))/(42 * 10 ↑ (-3))
```

que imprime: 5476191.6

b) Debe pulsarse:

```
PRINT (9.8 * 10 ↑ 14 + 5.876 * 10 ↑ 5)/(9.987 * 10 ↑ 5 - 10 ↑ 6)
```

que imprime: -7.5384728E+11

* * *

II.3. *Calcular*

$$x = \sqrt[7]{\frac{3 \text{ sen } 32^\circ 15'}{42{,}1^3}}$$

Solución

Debe pulsarse:

```
PRINT ((3 * SIN((32 + 15/6Ø) * π/18Ø))/(42.1 ↑ 3)) ↑ (1/7)
```

que imprime:

Ø.21531377

* * *

II.4. *Calcular: a) sen 48°, b) tag 73°*

Solución

a) Debe pulsarse:

```
PRINT SIN (48*π/18Ø)
```

que imprime:

Ø.74314483

b) Debe pulsarse:

```
PRINT TAN (73 * π/18Ø)
```

que imprime:

3.27∅8526

* * *

II.5. *Observar el comportamiento del ordenador al intentar calcular*

$$\sqrt{\frac{416{,}1^2-516{,}8^3}{6{,}31}}$$

Solución

Debe pulsarse

```
PRINT SQR((416.1 ↑ 2 - 516.8 ↑ 3)/6.31)
```

que da un mensaje de error, puesto que el radicando es negativo.

* * *

II.6. *¿Qué imprimiría un ordenador al pulsar*

PRINT 13 + ABS (1∅-4-15) ?

Solución

La expresión dada equivale a:

$$13 + |10-4-15| = 13+|-9| = 13 + 9 = 22$$

Por tanto imprimiría:

22

* * *

II.7. *¿Qué imprimiría un ordenador al pulsar*

PRINT SGN(16) ?

Solución

La expresión dada equivale al signo de 16, por tanto imprimiría 1, puesto que la función da:

$$\begin{cases} 1 & \text{si} \quad x > 0 \\ -1 & \text{si} \quad x < 0 \\ 0 & \text{si} \quad x = 0 \end{cases}$$

* * *

II.8. *¿Qué debería escribirse para que el ordenador nos dé un número natural aleatorio del 1 al 50 ?*

Solución

Debería pulsarse:

```
PRINT INT (RND * 5Ø) + 1
```

ya que la función RND nos da un número n: $0 \leqslant n < 1$
al multiplicar por 50 dará un número m: $0 \leqslant m < 50$
La función INT nos dará un número natural s: $0 \leqslant s < 50$
Al sumarle 1 dará un número t: $1 \leqslant t \leqslant 50$

* * *

EJERCICIOS PROPUESTOS

29. Explicar qué diferencia hay entre la memoria ROM y la memoria RAM.

30. ¿A qué se llaman memorias de masa?

31. Contestar a las siguientes preguntas:
a) ¿Cuál es el contenido de la memoria ROM?
b) ¿Se pueden almacenar datos en ella?

32. Al desconectar en ordenador, ¿se borra toda la memoria del mismo?

33. Cuando en informática se habla de "HARDWARE" se refiere:
a) A que el programador ha estudiado en HARVARD
b) Al soporte físico del ordenador
c) Al conjunto de programas que puede ejecutar el ordenador.
d) A la marca del ordenador.
Razonar la respuesta.

34. Calcular:
a) (1275,8+9876,56)-(471,9-2,456)
b) (987,4-876,34)-[(19830-98387456)+(198-34)]

35. Calcular:

a) $\dfrac{5,568095^5 \cdot 124,6}{1357,4^2}$ b) $\dfrac{4,9846^3 \cdot 987,65}{136,9^2+47183,8^3}$

36. Calcular:

a) $\sqrt{\dfrac{237,5 - 48,25}{(124,1)^2+7,8}}$ b) $\sqrt[3]{\dfrac{12,6^3 - 14}{24,5 - 3,4}}$

37. Calcular:

a) $(12300000000+2^{10}) \cdot \pi$

b) $(130000000-3^{12}) + 5^9$

38. Pasar a grados sexagesimales:

a) 41º 14' 24"

b) 125º 25' 45"

39. Comprobar que

$$\text{sen}^2 15º21'53'' + \cos^2 15º21'53'' = 1$$

40. Una ruedecilla gira uniformemente a 50 revoluciones por minuto, su radio mide R=3,15 cm. Calcular:

a) El periodo (T = tiempo que tarda en dar una vuelta)

b) La pulsación ($\omega = 2\pi/T$)

c) La frecuencia ($N = 1/T$)

d) La velocidad lineal ($v = \omega R$)

e) La aceleración centrípeta ($a_n = v^2/R$)

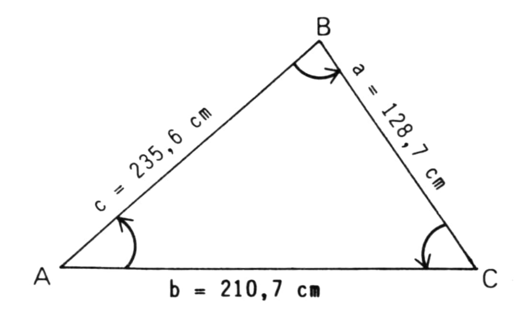
41. Hallar el ángulo a del primer cuadrante tal que:

a) sen a = 0,7543
b) cos a = 0,3427
c) tag a = 2,4896

42. Hallar:
a) sen $(8\pi/7)$
b) tag $(6\pi/5)$

43. Hallar el angulo a del primer cuadrante tal que:
a) sen a = 0,9835476
b) tag a = 5,475908
c) cos a = 0,4857

44. Dado el triángulo $\overset{\triangle}{ABC}$ del gráfico:



comprobar que $\hat{A}+\hat{B}+\hat{C} = 180^{o}$ aplicando el teorema del coseno $(a^2 = b^2+c^2-2bc \cos \hat{A})$.

45. Calcular: a) π^e ; b) e^π

46. Hallar: a) L4,5 ; b) L(54,8.9,87)

47. Hallar: a) $\log_{10} 2$; b) $\log_{10}(45,8\cdot\pi)$

48. Sabiendo que el límite de la sucesión de término general

$$a_n = \left(\frac{n+1}{n+3}\right)^{2n} \quad \text{es} \quad e^{-4} ,$$

hallar la diferencia entre el límite y el término 23 de dicha sucesión.

49. Calcular:

$$L\left(\frac{10^{-2}+5,86}{4^3-10^{-3}}\right)$$

50. ¿Qué imprimirá el ordenador al pulsar:

```
PRINT ((-3) ↑ 2 + 7 * 5.7) ↑ (1/7) ?
```

51. Calcular:
a) $(-35)^4$
b) $(-1,845)^{12}$

52. Intentar hallar tag 90º, ¿qué le pasa al ordenador? Dar una explicación.

53. Intentar hallar: log(-10), ¿qué le pasa al ordenador? Dar una explicación.

54. ¿Qué imprimiría un ordenador al pulsar:
a) PRINT (-54+3ØØ)/3*5
b) PRINT 5ØØ/8/4*1Ø ?

55. ¿Qué imprimiría un ordenador al pulsar:
a) PRINT 3Ø*(4+16Ø)/6+3*5
b) PRINT 4Ø/2Ø+1Ø*5-18 ?

56. ¿Qué hay que escribir para que un ordenador simule una lotería del 1 al 100?

57. ¿Cuál es el valor máximo que puede tomar la expresión

INT(RND*1ØØØ)+1 ?

58. ¿Entre qué números estaría el resultado de pulsar PRINT RND*1ØØ ?, ¿y de pulsar PRINT (RND*1Ø)+1 ?

3
DIAGRAMA DE FLUJO. INICIACION A LA PROGRAMACION

1. Diagrama de flujo

Definición 1

Llamaremos **Algoritmo** al conjunto de acciones destinadas a la resolución paso a paso de un problema.

Se caracteriza por tener un número finito de pasos bien definidos o sea sin ambigüedad.

Ejemplo

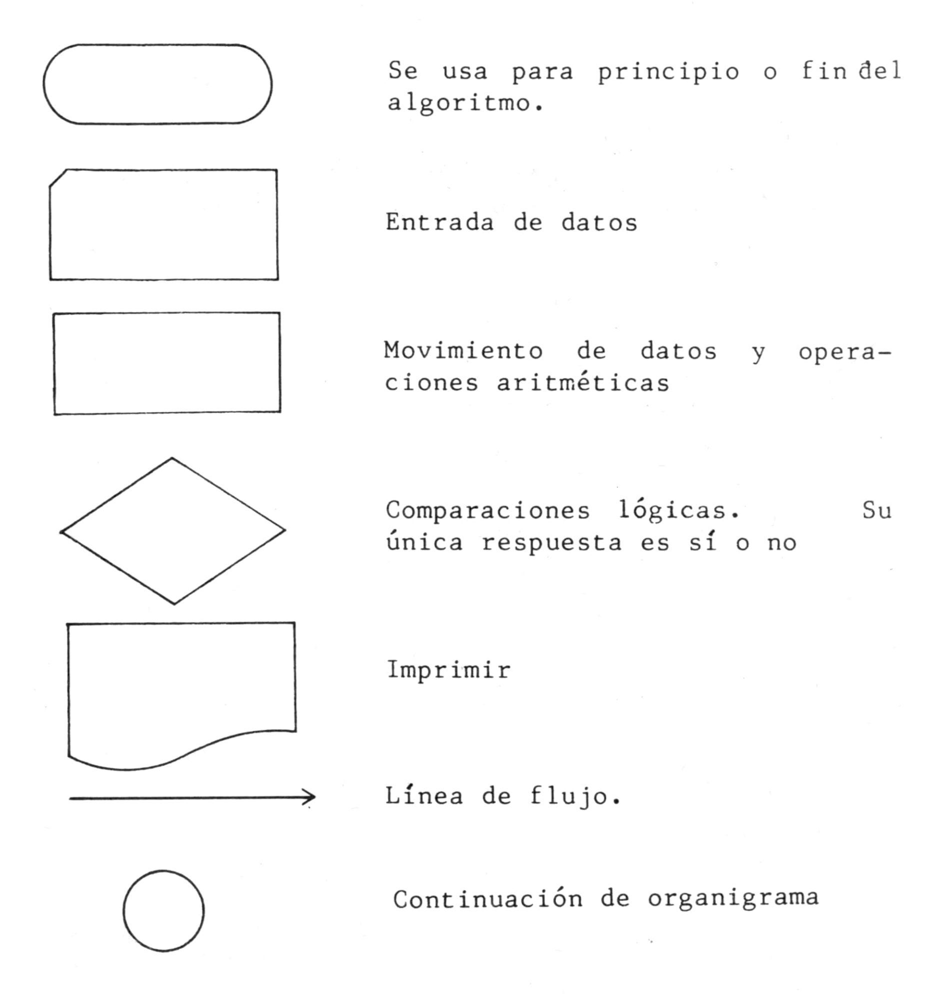
El problema de: "Poner el coche en marcha", puede resolverse con el siguiente algoritmo:

1. Abrir el coche.
2. Poner la llave de contacto.
3. Girar la llave.
4. ¿Se pone el motor en marcha?. En caso afirmativo fin del problema; en caso negativo ir a 5.
5. Mirar si la batería está cargada. En caso afirmativo ir a 3; en caso negativo llamar al mecánico y fin del problema.

Definición 2

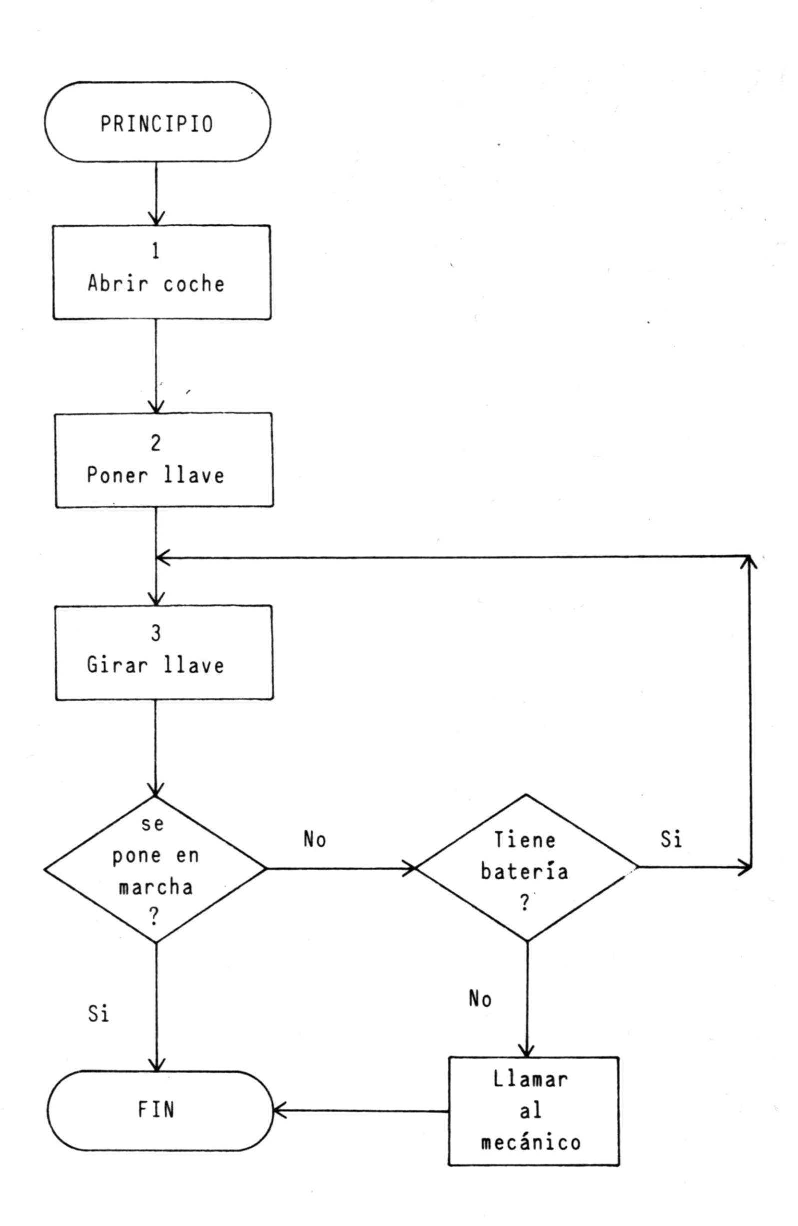
Llamaremos **Diagrama de flujo** u **Organigrama** al esquema que usamos para representar un algoritmo.

Los símbolos



Ejemplos

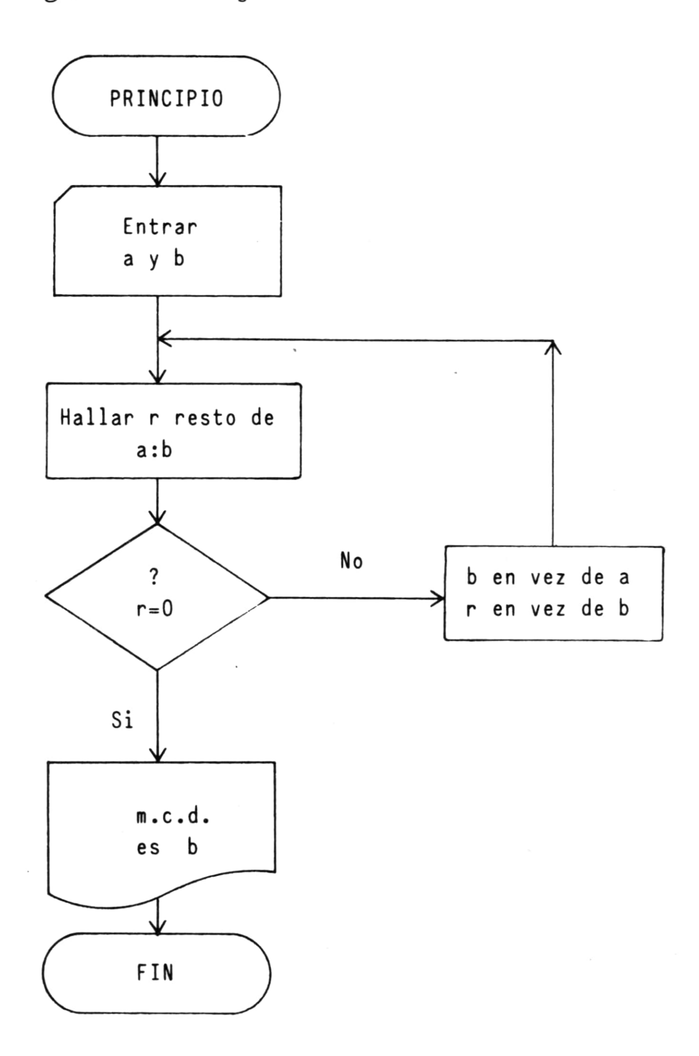
a) El organigrama correspondiente al ejemplo anterior es:



b) El cálculo del m.c.d. de dos números naturales a y b por el algoritmo de Euclides puede realizarse así:

1. Entrar a y b
2. Hallar el resto de a:b (r)
3. ¿Es cero el resto?.
   Si es así escribir: m.c.d. es b y fin del programa
   En caso contrario ir a 4
4. Colocar b en vez de a y r en vez de b. Ir a 2

El diagrama de flujo es:



También puede utilizarse el algoritmo basado en la siguiente propiedad:

Si $a > b \Rightarrow$ m.c.d.(a,b) = m.c.d.(a-b,b)

Definición 3.

Si el algoritmo lo escribimos para que lo ejecute un ordenador decimos que hemos realizado un "**programa**" para el mismo.

Es decir un **programa** consiste en una serie de órdenes numeradas que el ordenador efectúa en riguroso orden de numeración.

Resumiendo

Para que un ordenador nos resuelva un problema seguiremos los siguientes pasos

1 Análisis del problema.
2 Obtención de un algoritmo para su resolución.
3 Organigrama del algoritmo.
4 Hacer el programa correspondiente.

Nótese que un mismo problema puede resolverse mediante algoritmos distintos.

2. Lenguajes de programación

En un principio los programas debían hacerse utilizando el llamado "**Lenguaje máquina**" basado en el "Código binario", sin embargo esto implicaba un

conocimiento interno del ordenador que generalmente es muy complicado. Por esto se crearon unos lenguajes de programación más cercanos al lenguaje natural para podernos comunicar con el ordenador. Estos lenguajes se llaman de **"alto nivel"**. Nosotros veremos una introducción a uno de estos lenguajes: el BASIC (Beninger's All-purpose Symbolic Instruction Code: Código de Instrucciones de uso general para principiantes) nacido en la década de los 60.

Para que el ordenador entienda los lenguajes de programación ya sea el BASIC u otro de alto nivel necesita que se lo traduzcan. El encargado de hacerlo es un **compilador** o un **intérprete** (situado en la memoria ROM), según que la traducción del programa se realice toda de una vez o bien línea por línea hasta acabar todo el programa.

Un programa escrito en lenguaje de alto nivel se llama **programa fuente** y una vez traducido al "lenguaje máquina" se llama **programa objeto.**

Nosotros no nos ocuparemos de la forma de traducción. Trataremos solamente de comunicarnos con el ordenador sabiendo que si no cometemos errores en el lenguaje que usamos, el ordenador lo asimilará gracias a esta "traducción simultánea" de la que no nos preocuparemos.

Antes de continuar tenemos que advertir que bajo la denominación de **lenguaje** BASIC hay una serie de "dialectos" que en lo básico funcionan igual, pero que los distintos modelos de ordenadores hacen que surjan en algunos detalles pequeñas diferencias. El dominio de una forma de programar, allana las dificultades que puedan surgir para pasar a otro lenguaje o bien a otro dialecto.

En el Tema 2 ya hemos empleado la forma de operar y las funciones en BASIC.

A partir de ahora explicaremos la sintaxis propia de este lenguaje para poder programar.

3. Variables

Cuando en Matemáticas hablamos de variables, nos referimos a letras que representan números.

En lenguaje BASIC pueden representar números (**variables numéricas**) o bien una serie de caracteres: letras, espacios en blanco y otros signos (**variables alfanuméricas**).

Las **variables numéricas** se representan con una letra: A,B,... o bien una letra seguida de otras o números.

Por ejemplo:

A , B , C , H1 , P2 , PALO

Algunos ordenadores sólo distinguen los dos primeros caracteres, por ejemplo para este tipo de ordenadores es la misma variable PA que PALO.

Si una variable numérica sólo ha de tomar valores enteros se escribe el nombre de la variable seguida del símbolo %.

Ejemplo: A% ; B2% ; ABC%

Con ello se consigue un ahorro de memoria.

Las **variables alfanuméricas** se caracterizan por tener el caracter $ al final de la variable.

Por ejemplo:

A$, B2$, PALO$

Algunos ordenadores solamente admiten una letra delante del signo $.

El valor que les atribuímos son **cadenas de caracteres**, por ejemplo:

A$ = "CURSO DE INFORMATICA"

El valor de la variable debe ir entre comillas.

Con las variables numéricas pueden efectuarse las operaciones indicadas en el tema anterior. En cambio con las alfanuméricas sólo se puede realizar la "operación" suma entendida como yuxtaposición de dos cadenas.

Ejemplo

- "IN" + "VERTEBRADO" = "INVERTEBRADO"

- "ESCUCHO"+" "+"MUSICA" = "ESCUCHO MUSICA"

- "124"+"1Ø" = "1241Ø"

Obsérvese que "124", "1Ø" y "1241Ø", son cadenas, no representan pues cantidades numéricas para efectuar operaciones aritméticas.

Una cadena puede tener a lo sumo 255 caracteres (hay alguna excepción).

Nota. No puede usarse como variable palabras reservadas para el lenguaje del ordenador, por ejemplo: SQR, SIN, PRINT, ...

Podemos imaginar las variables como unos cajones que guardan una cantidad (variable numérica) o una cadena de caracteres (variable alfanumérica)

4. Sentencia LET (Sea)

Con la sentencia LET, asignamos un valor a una variable directamente o a través de una función. Por ejemplo:

LET X = 7 ; LET Y = X * X - 5 * X + 8

Así pues, LET X = 5, nos indica que cuando aparezca la X en nuestro programa toma el valor 5.

Si se le asigna un nuevo valor a una variable ya definida, este valor desplaza al antiguo.



Esta sentencia no es siempre necesaria, pues en algunos ordenadores para asignar valores a variables se puede omitir el LET.

Por ejemplo:

X = 5

El signo igual (=) no siempre coincide con el concepto matemático del mismo, pues si escribimos:

LET X = X + 1

no quiere decir que tenemos una ecuación, sino que significa: "el valor de la variable X lo cambiamos por el valor X+1". Es decir, que si antes era X=5, despues de LET X = X+1 será X=6.

Resumiendo

> Con la sentencia **LET** le decimos al ordenador, "asigna a la variable de la izquierda del signo = el valor de la expresión de la derecha".

Ejemplo

```
LET X = A + B
```

significa pues que el valor de la variable X es suma de los valores de las variables A y B.

Si las variables son alfanuméricas, al atribuirles un valor lo ponemos entre comillas, es decir, si queremos que la variable A$ sea igual a NOMBRE, B$ sea PRIMER APELLIDO y C$ sea SEGUNDO APELLIDO, escribiremos:

```
LET A$ = "NOMBRE"
LET B$ = "PRIMER APELLIDO"
LET C$ = "SEGUNDO APELLIDO"
```

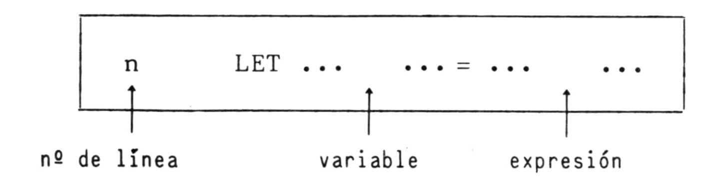
Nota. Dejar un espacio en blanco entra también como un caracter más en el valor de la variable. Así por ejemplo son distintos:

```
LET A$ = "INTER ROGATIVO"
```

y

```
LET A$ = "INTERROGATIVO"
```

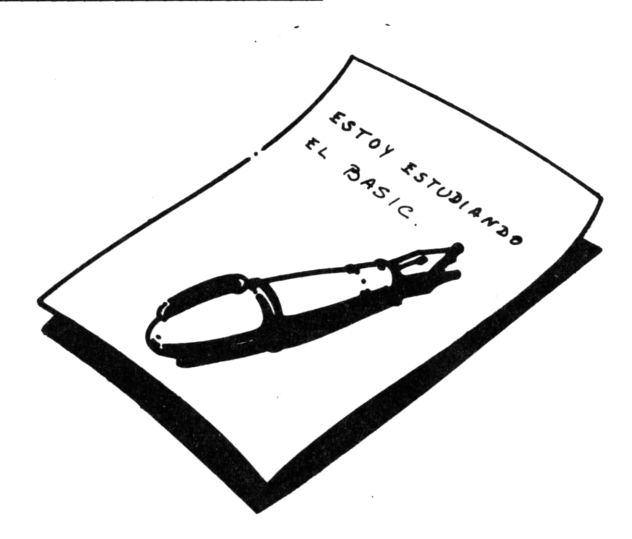
Las instrucciones que contienen esta sentencia corresponden al siguiente formato:



Nota. Si a una variable no se le asigna ningún valor, generalmente el ordenador entiende que su valor es Ø para las variables numéricas o bien la cadena vacía para las alfanuméricas.

Hay algún ordenador que no admite el uso de variables si antes no se les ha asignado algún valor dando en este caso el mensaje "variable no definida".

5. Sentencia PRINT (Imprimir)



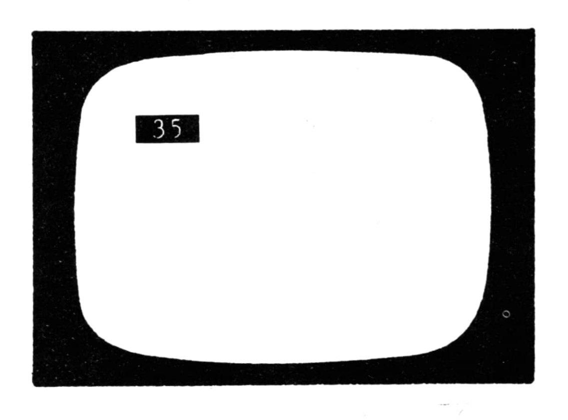
Con la sentencia PRINT escribimos en la pantalla la expresión que nosotros deseamos. La hemos utilizado ya al hablar del ordenador como calculadora.

Ejemplo

Pulsando

PRINT 5*7

aparece en pantalla (después de pulsar RETURN o ENTER, ...):



Si queremos que imprima un mensaje (cadena) nos basta poner el mismo entre comillas.

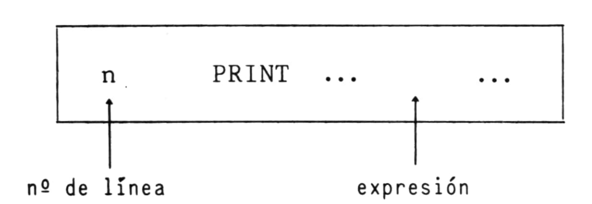
Ejemplo:

PRINT "ESTOY ESTUDIANDO EL BASIC"

nos da en pantalla:



Las instrucciones que contienen esta sentencia corresponden al siguiente formato:



También se usa PRINT para imprimir valores de variables. Por ejemplo PRINT X imprime el valor de la variable X ya definida. Análogamente PRINT A$ imprime la cadena asignada a la variable A$.

Ejemplo

- Si a la variable X le asignamos el valor 4

PRINT X

imprimirá:

- Si el valor de X es 4 y el de Y es 5

PRINT X+Y

imprimirá:

9

- Si A$ = "ROSA" y B$ = "MARI"

PRINT A$+B$

imprimirá:

ROSAMARI

6. Presentación en pantalla

a) Uso de la coma en la sentencia PRINT

La pantalla está dividida en zonas a las que se accede a través de la coma (,). El número de zonas depende del modelo de ordenador.

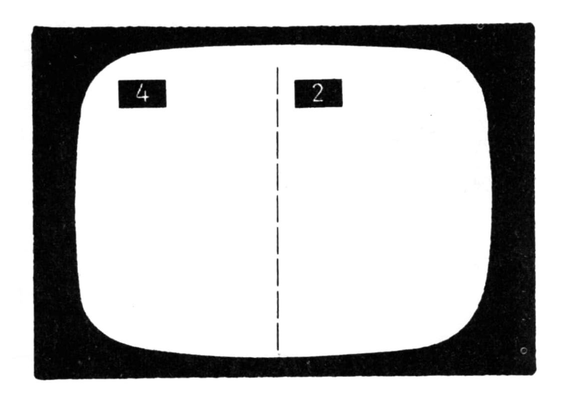
Algunos ordenadores dividen la pantalla en dos zonas, otras en tres, cuatro o más.

Evidentemente cada zona admite un número máximo de caracteres.

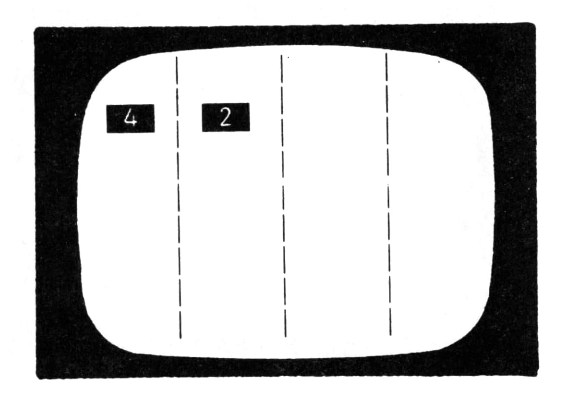
Cada vez que ponemos la coma en la sentencia PRINT capta la orden de desplazarse por las zonas que tiene. Si por ejemplo ponemos:

PRINT 4,2

obtenemos en pantalla:



ordenador con 2 zonas

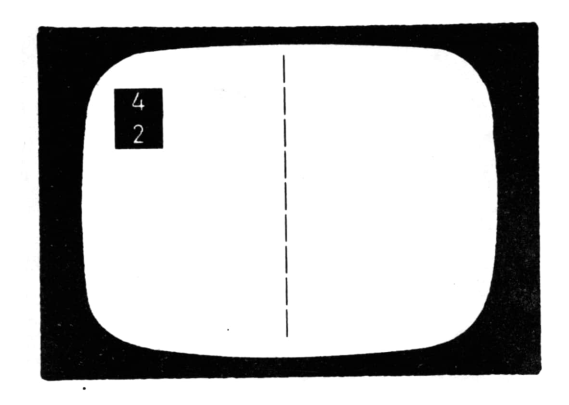


ordenador con 4 zonas

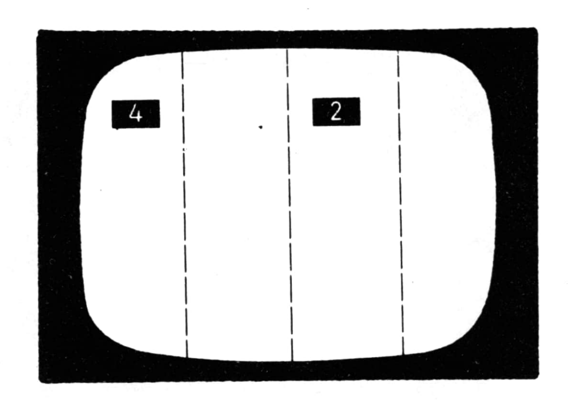
Si pulsamos

PRINT 4 , , 2

obtenemos en pantalla:



ordenador con 2 zonas



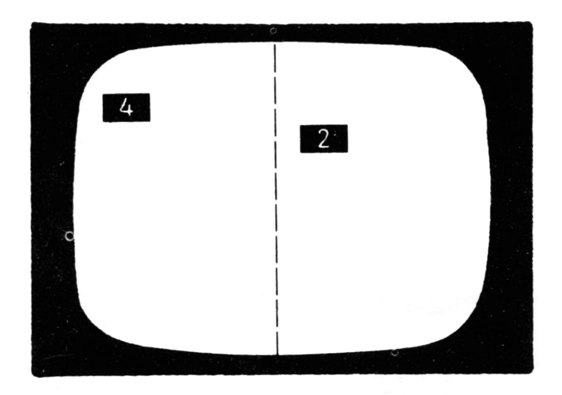
ordenador con 4 zonas

Obsérvese que la segunda coma desplaza el 2 hacia el tercer lugar, que en caso de existir sólo dos zonas es otra vez el inicial y por consiguiente cambia de línea:

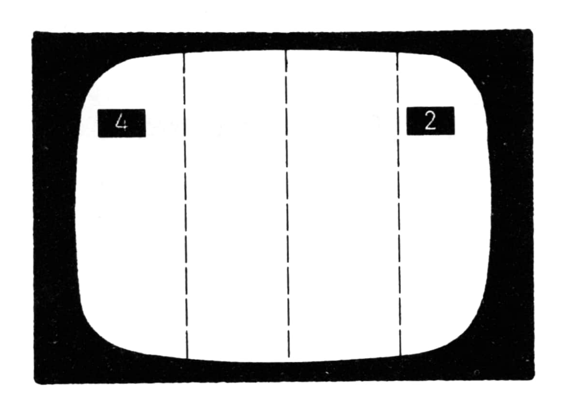
Si pulsamos

PRINT 4 , , , 2

obtenemos en pantalla:



ordenador con 2 zonas



ordenador con 4 zonas

Así sucesivamente.

b) Uso del punto y coma en la sentencia PRINT

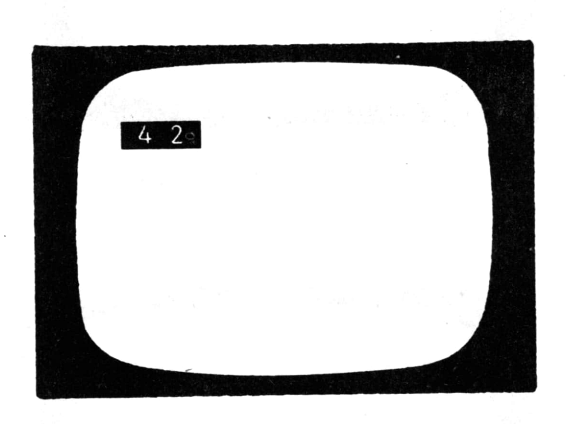
Si ponemos un punto y coma (;) entre dos números se imprimen uno a continuación del otro dejando un espacio en blanco. (Algún ordenador omite este espacio en blanco)

Ejemplo:

Si pulsamos

PRINT 4 ; 2

obtenemos:



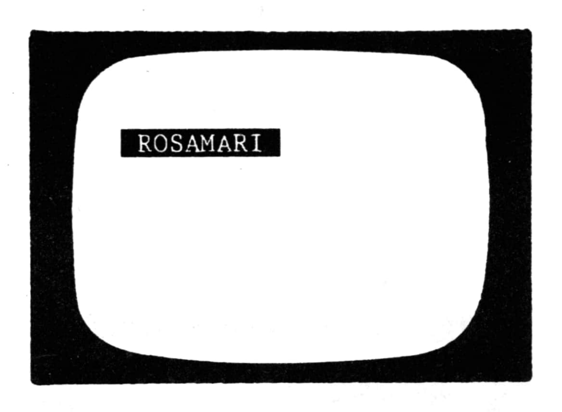
Se pueden hacer combinaciones de comas y puntos y comas para desplazar el resultado según convenga.

Si ponemos un punto y coma (;) entre dos cadenas se imprimen una a continuación de la otra (sin espacio en blanco)

Por ejemplo si pulsamos

PRINT "ROSA" ; "MARI"

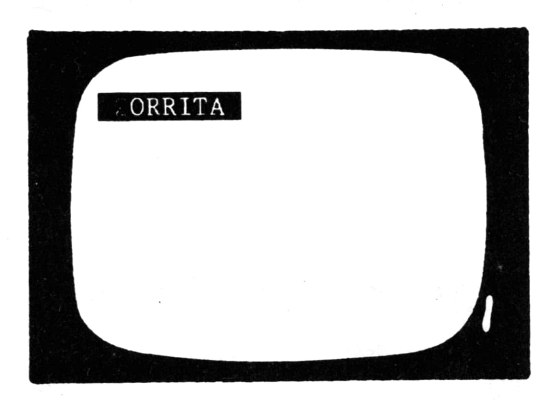
aparece en pantalla:



Si pulsamos

PRINT "ZOR" ; "RITA"

aparece en pantalla:



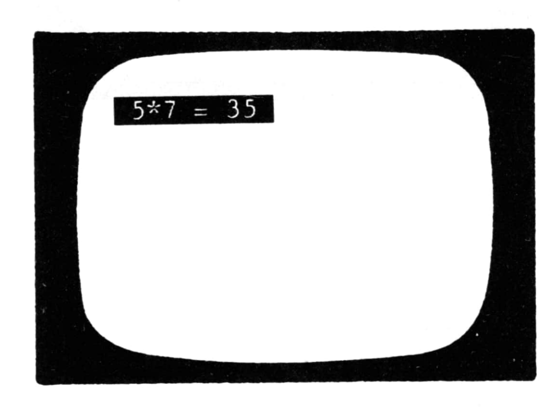
También podemos imprimir un mensaje y el resultado de una operación aritmética; para ello tenemos que recordar que el mensaje va entre comillas, no el resultado de la operación.

Ejemplo

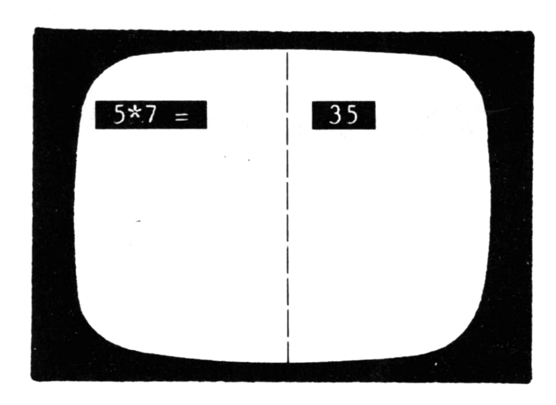
Si queremos que en pantalla aparezca 5*7 = 35 nos basta pulsar:

PRINT "5*7 =" ; 5*7

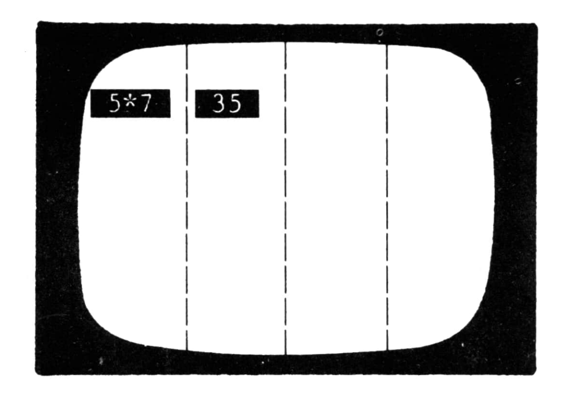
que imprime en la pantalla:



Si en vez de poner punto y coma (;) ponemos una coma (,) el resultado de pulsar PRINT "5*7 =" , 5*7 es:



Ordenador con 2 zonas



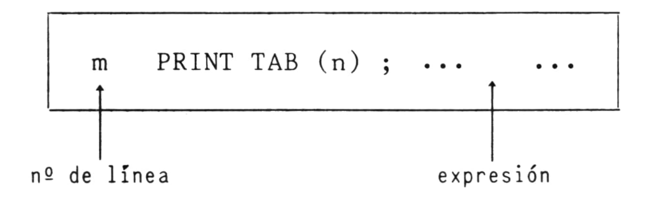
Ordenador con 4 zonas

es decir, no sale 35 junto al signo igual.

c) Uso de TAB

El ordenador divide la pantalla en columnas cuyo número depende del modelo.

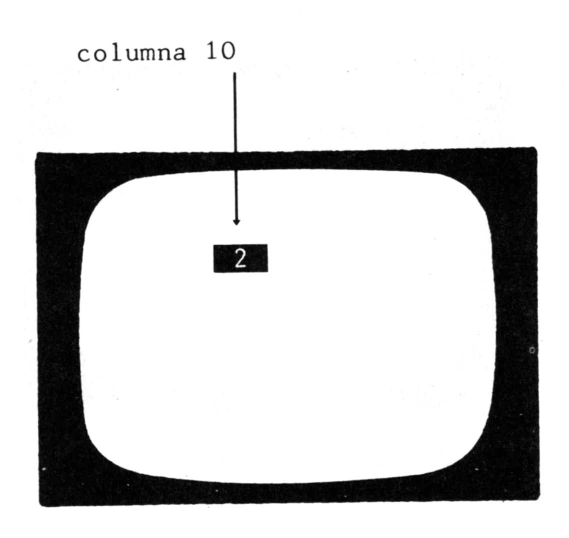
Se empieza a imprimir en la columna n (n menor o igual que el número de columnas) con la instrucción de formato:



Ejemplo

PRINT TAB(1Ø) ; SQR(4)

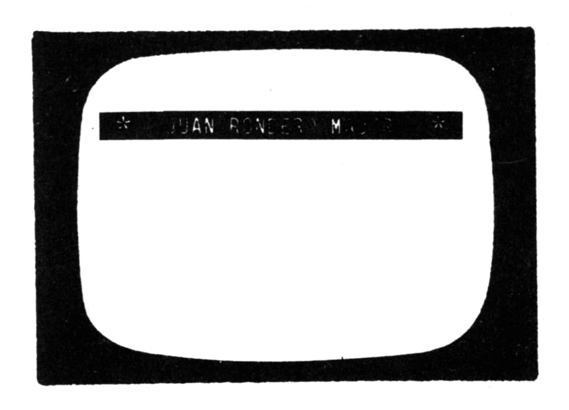
imprime:



Ejemplo:

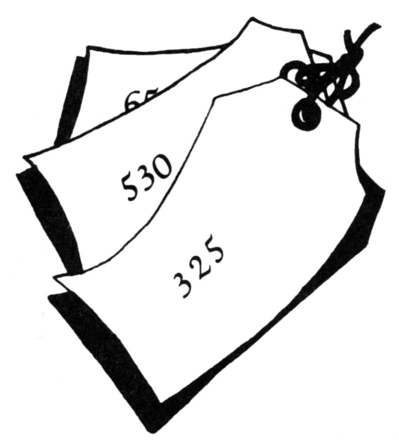
```
PRINT TAB(2);"*";TAB(6);"JUAN RONDERO MAJOR";TAB(29);"*"
```

imprime:



7. Numeración de las instrucciones

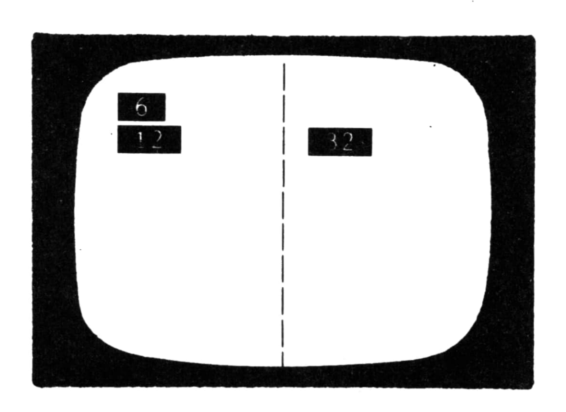
Para que el ordenador sepa cómo efectuar las instrucciones que componen un programa, tenemos que numerarlas y las efectuará en riguroso orden de lista.



Dicha numeración va desde 1 hasta 9999. El programa

```
1  PRINT 4+2
2  LET X = 4
3  LET Y = 8
4  PRINT X + Y , X * Y
```

nos daría en pantalla (**después de pulsar RUN**):

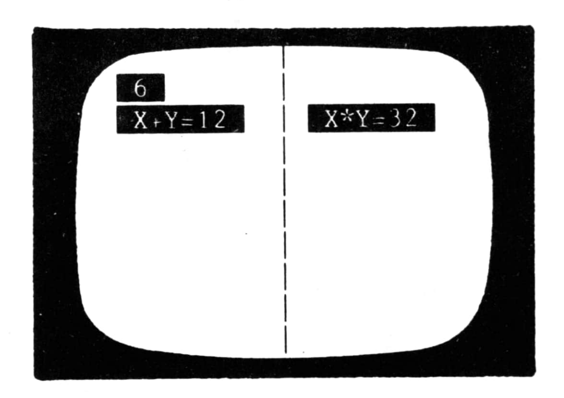


Ordenador con 2 zonas

El resultado que aparece, no nos indicará de qué operaciones proviene, si lo quisiéramos nos bastaría cambiar la instrucción 4 por la siguiente

```
4 PRINT "X+Y=";X+Y,"X*Y=";X*Y
```

con lo cual tendríamos en la pantalla:



Ordenador con 2 zonas

En muchas ocasiones al efectuar un programa nos encontramos que queremos intercalar una instrucción entre dos ya escritas, lo cual nos ocasiona un problema si son correlativas. No podemos intercalar una línea entre la 3 y la 4. Es fácil vencer esta dificultad. Para ello se suele poner la numeración de 10 en 10 y así dejamos espacio entre dos líneas para poder intercalar otras si fuera necesario.

El programa anterior lo escribiremos así:

```
10 PRINT 4+2
20 LET X = 4
30 LET Y = 8
40 PRINT "X+Y=";X+Y,"X*Y=";X*Y
```

Si intercalásemos ahora la línea

```
35 PRINT "X-Y=";X-Y,"X/Y=";X/Y
```

el ordenador la colocaría entre la 3Ø y la 4Ø aunque nosotros la hemos escrito después. El resultado de este programa es:

```
6
X-Y = -4        X/Y = Ø.5
X+Y = 12        X*Y = 32
```

Si es necesario rectificar una línea puede escribirse de nuevo y ésta reemplaza a la antigua.

Este método obliga a escribir de nuevo toda la línea rectificada, sin embargo existen otras formas de modificar sólo una parte. Estas formas dependen del modelo del ordenador. Se accede a ellos mediante el comando EDIT.

Si queremos suprimir una línea basta escribir el número de orden de dicha línea y pulsar RETURN o ENTER.

Nota: Obsérvese que siendo X,Y variables numéricas, al poner "X-Y" no efectúa esta operación, pues para el ordenador lo entrecomillado es un mensaje para escribirlo, no para efectuar operaciones.

8. Comando RUN (correr)

Según hemos visto en la pregunta anterior, para poder ejecutar un programa (lo que se llama correr un programa) utilizaremos el comando RUN. Este comando borra todas las variables e inicia el programa desde su primera línea.



Si en un programa hay varias partes independientes que forman, podríamos decir como un subprograma del mismo, como el siguiente:

```
10  ... ...  }
20  ... ...  }
:
:
70  ... ...  }
80  ... ...  }
90  ... ...  }
:
:
150 ... ...  }
:
:
```

y a nosotros sólo nos interesa una de las partes, por ejemplo la segunda, nos bastará con poner

```
RUN  80
```

con lo que ignorará las líneas anteriores y empezará por la 80.

Ejemplo

En el siguiente programa

```
10  PRINT 4
20  LET X = 5
30  LET Y = 6
40  PRINT X,Y
```

al pulsar RUN obtenemos en pantalla:

```
4
5          6
```

En cambio, al pulsar

```
RUN  20
```

obtenemos:

5 6

y al pulsar

RUN 3Ø

dará o bien

mensaje de error puesto que en la línea 4Ø aparece la variable X que nos estaba definida por haber empezado después de la 2Ø

o bien

Ø 6

si el ordenador asigna el valor Ø a las variables no definidas.

El comando RUN puede estar incluído en una línea de programa.

9. Sentencia END (fin)

Con la sentencia END indicamos al ordenador el final del programa, aunque para la ejecución del mismo no es necesario pulsarla.



En el programa anterior se podría acabar añadiendo la línea

5Ø END

En los programas que escribiremos en este libro haremos uso de esta sentencia.

10. Línea, instrucción, sentencia (orden) y comando

En este capítulo hemos utilizado los términos: línea, instrucción, sentencia (orden) y comando.

No hay uniformidad en la interpretación de estas palabras. Adoptaremos en este libro los siguientes criterios.

Un programa está formado por líneas numeradas.

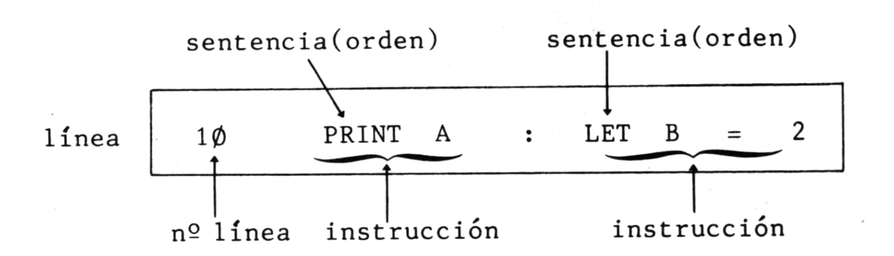
Cada línea está formada por una o más instrucciones separadas por dos puntos (:).

Cada instrucción está formada por una sentencia (orden) aplicada a una expresión.

Las sentencias que pueden usarse sin formar parte de un programa las llamaremos comandos.

Son comandos: PRINT, RUN, LET, ...

Ejemplo



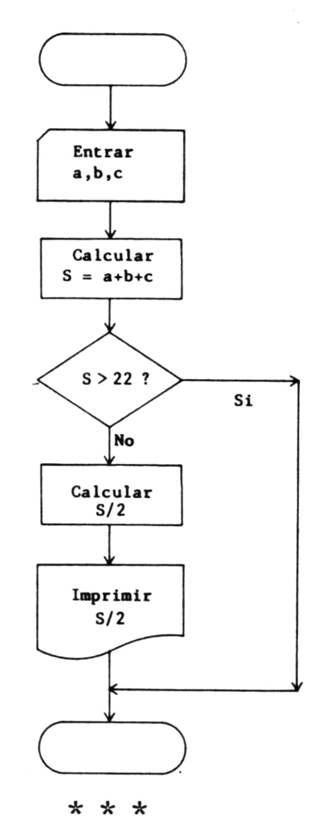
EJERCICIOS RESUELTOS

III.1. *Hacer el organigrama del siguiente algoritmo:*

1. *Entrar tres números*
2. *Sumarlos*
3. *Si la suma es mayor que 22 ir a 7*
4. *Si la suma es menor o igual que 22 ir a 5.*
5. *Dividir la suma por 2.*
6. *Imprimir este resultado*
7. *Fin de programa*

Solución

El organigrama es:

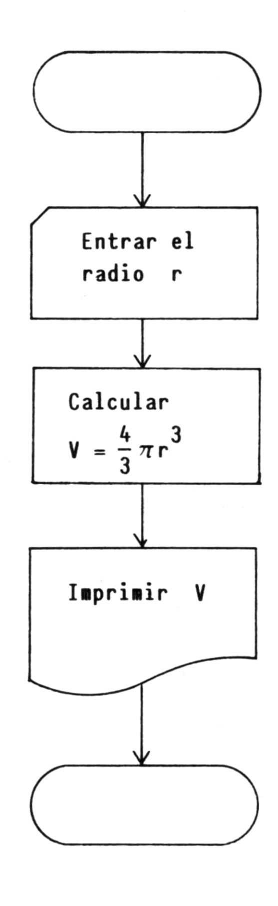


* * *

III.2. *Hacer el organigrama de un algoritmo correspondiente al problema: Hallar el volumen de una esfera ($V = 4/3\,\pi r^3$), con las siguientes acciones*
1. Entrada del radio
2. Efectuar las operaciones
3. Imprimir el resultado

Solución

El organigrama es:

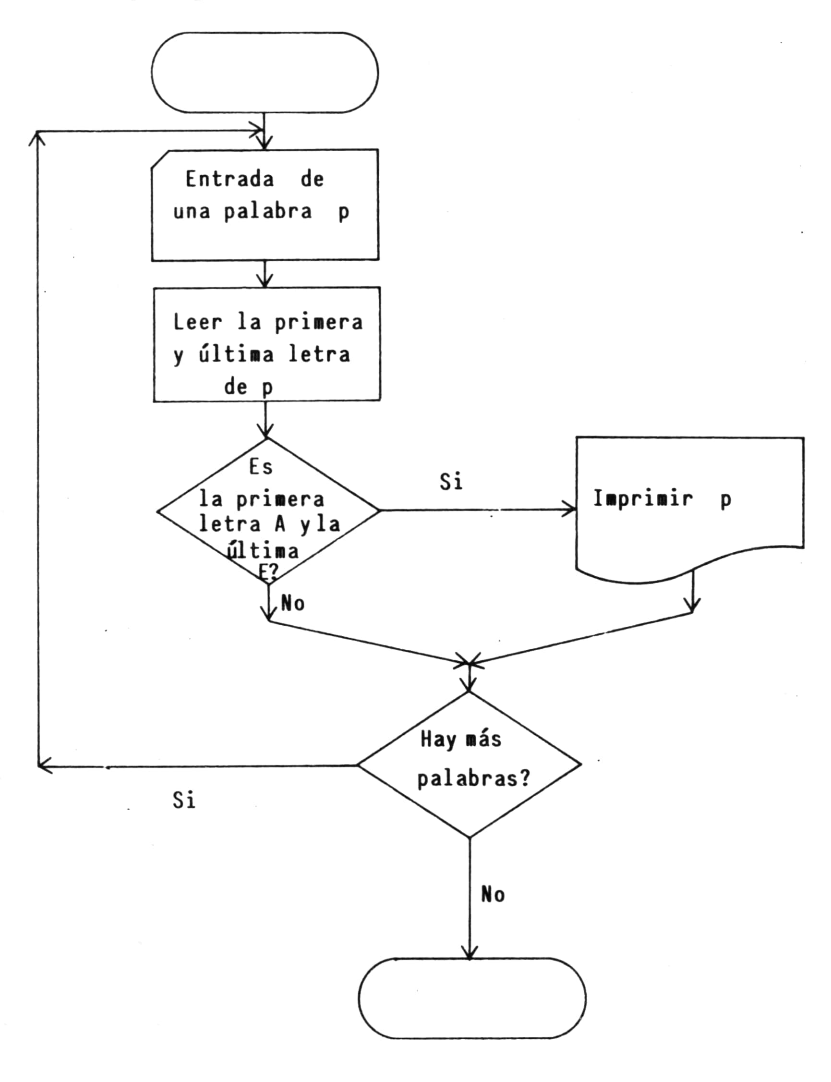


* * *

III.3. *Hacer el organigrama de un algoritmo correspondiente al problema: Hallar las palabras que empiecen por A y acaben en E de entre una lista de palabras. Imprimir estas palabras*

Solución

El organigrama es:

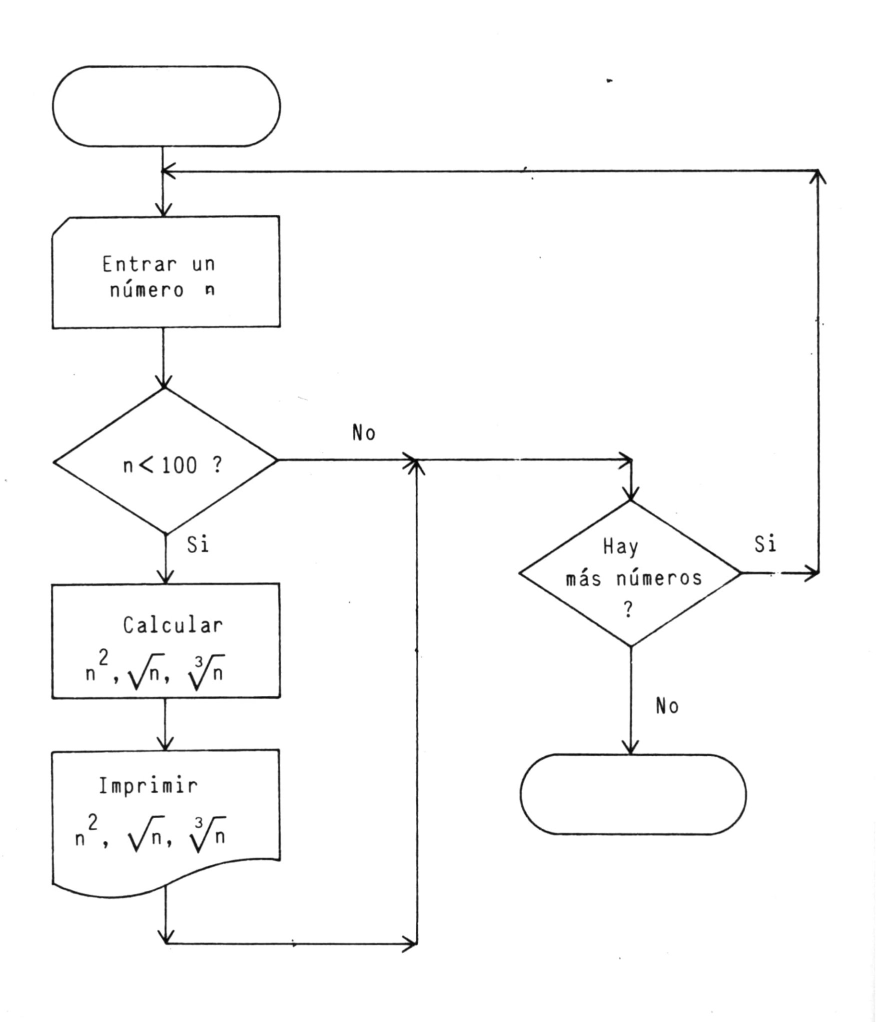


* * *

III.4. *Hacer el organigrama de un algoritmo correspondiente al siguiente problema: Dada una serie de números positivos, si son menores que 100 imprimirlos y hallar e imprimir su cuadrado, su raíz cuadrada y su raíz cúbica.*

Solución

El organigrama es:

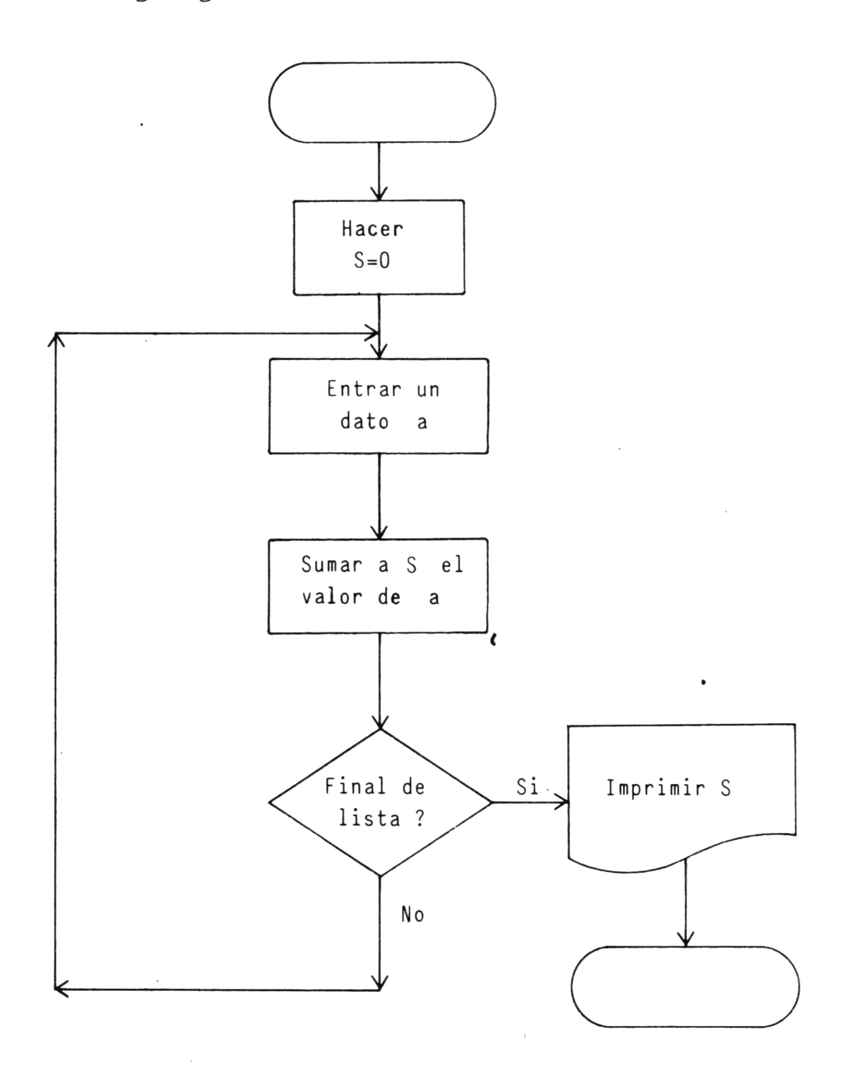


* * *

III.5. *Hacer el organigrama de un algoritmo correspon- al problema: Dada una lista de números, calcular la suma de todos ellos.*

Solución

El organigrama es:



* * *

III.6. *¿Cuáles de las siguientes expresiones se pueden utilizar como variables?. Razonar la respuesta.*
a) X1; b) 1X; c) XY; d) HH; e) 5Z

Solución

Son válidas como variables:

a), c) y d)

y no son válidas las b) y e) por empezar por un número en vez de una letra.

* * *

III.7. *Explicar qué es un error de sintaxis.*

Solución

Se comete un error de sintaxis si al escribir un programa en un lenguaje de alto nivel (como el BASIC) se infringen las reglas (sintaxis) de este lenguaje.

* * *

III.8. *Escribir en forma de sentencias LET:*

a) $b = \dfrac{(a+b-c)^2}{3a-1}$; b) $h = \dfrac{m^k-3m}{k+2^4}$

Solución

Dichas sentencias son:

a) LET b = (a+b-c)↑2/(3*a-1)

b) LET h = (m↑k-3*m)/(k+2↑4)

* * *

III.9. *¿Cuál es el resultado de la ejecución del siguiente programa?*

```
10 LET X = 20
20 LET Y = 40
30 PRINT X+Y
40 LET X = X+1
50 LET Y = Y+2
60 PRINT X*Y
70 LET X = X+2
80 PRINT X-Y
90 END
```

Solución:

En la 30 suma y el resultado es 60.

En la 40 y 50 el valor de las variables es X=21, Y=42 con lo cual la 60 nos da como resultado 882.

En la 70 el valor de X es 23 por lo que el resultado de la 80 es -19

La pantalla nos da pues:

```
60
882
-19
```

* * *

III.10. *¿Que aparecerá en pantalla con la ejecución*

del siguiente programa?:

```
10 LET A$ = "Juan Angel" : LET B$ = "Rosi" :
   LET C$ = "Mariano"
20 LET X=2 : LET Y=4
30 PRINT A$; " estudia ";X;" horas cada día, mien-
   tras que ";C$;" y ";B$;" tienen bastante con ";
   Y;" horas a la semana"
40 END
```

Solución

Aparecerá:

Juan Angel estudia 2 horas cada día, mientras que Mariano y Rosi tienen bastante con 4 horas a la semana.

* * *

III.11. *Si introdujéramos el siguiente programa:*

```
40 PRINT A*B
20 PRINT A-B
 5 LET A=20
10 LET B=4
50 END
```

en un ordenador ¿daría mensaje de error? En caso negativo qué imprimiría?

Solución

No hay errores. Dicho programa nos imprime como resultado:

```
16
80
```

* * *

III.12. *¿Qué imprimirá la ejecución del siguiente programa?*

```
10 LET A$ = "EX"
20 LET B$ = "ALUMNO"
30 PRINT A$+B$
40 END
```

Solución

Dicho programa imprimirá:

EXALUMNO

* * *

III.13. *¿Hay algún error de sintaxis en el siguiente programa?*

```
10 LET X=12,16
20 LET Y=4,5
30 PRINT X+Y
40 PRINT X·Y
50 END
```

Si lo hay señalarlo, y si no lo hay decir cuál sería el resultado de la ejecución del mismo.

Solución

Hay tres errores. En las líneas 10 y 20 a las variables se les asigna valores con una coma y en BASIC la parte entera se separa de la decimal por un punto.

Por otra parte en la línea 40 el signo de multiplicar debería ponerse con un asterisco o sea:

X*Y

* * *

III.14. *Si la pantalla de un ordenador tiene 32 columnas, escribir una línea de programa para que imprima centrado en la pantalla el texto "SOLUCIONES"*

Solución

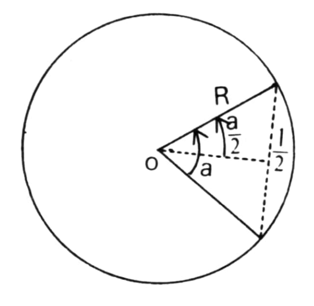
```
10 PRINT TAB(16-5); "SOLUCIONES"
```

Obsérvese que 16 es el número de columnas dividido por 2 y que 5 es el número de caracteres de la palabra "SOLUCIONES" dividido por 2.

* * *

III.15. *Hacer un programa en el que aparezcan en pantalla:*
a) La longitud del lado del polígono regular de 127 lados inscrito en una circunferencia de radio 2 cm.
b) La diferencia entre su perímetro y el de la circunferencia .

Solución



Se tiene que:

$$R \text{ sen } \frac{a}{2} = \frac{l}{2} \Rightarrow l = 2R\text{sen } \frac{a}{2} \quad ,$$

siendo $a = 2\pi/n$ (n es el número de lados del polígono regular).

El programa es:

```
10 LET n=127
20 LET a=2*π/n
30 LET R=2
40 PRINT "Lado= ";2*R*SIN(a/2)
50 PRINT
60 PRINT
70 PRINT "Diferencia= ";2*R*SIN(a/2)*n-2*π*R
80 END
```

* * *

III.16. *Hacer un programa en el que el ordenador compruebe la desigualdad*

$$\frac{a+b}{2} \geqslant \sqrt{a.b}$$

sindo a,b dos números entre 0 y 100 escogidos al azar por el ordenador.

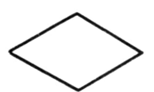
Solución

El programa es:

```
10 LET a = RND*100
20 LET b = RND*100
30 PRINT (a+b)/2,(a*b)↑0.5
40 PRINT "La diferencia entre el primero y
   el segundo es  ";(a+b)/2-(a*b)↑0.5
50 END
```

* * *

EJERCICIOS PROPUESTOS

* 59. Cuando en un organigrama aparece el símbolo  nos indica que:

a) Calculemos el área del rombo.
b) Hay una entrada-salida de datos.
c) Hay una comparación lógica cuyo resultado solamente es: si o no.
d) Hay una comparación lógica cuyo resultado es: algunas veces si, otras veces no.

Razonar la respuesta correcta.

60. Hacer el organigrama de un algoritmo correspondiente al siguiente problema: HALLAR LA SOLUCION DE UNA ECUACION DE PRIMER GRADO ($ax = b$, $a \neq 0$).

61. Hacer el organigrama del algoritmo de ENTRAR EN CLASE, con las siguientes acciones:

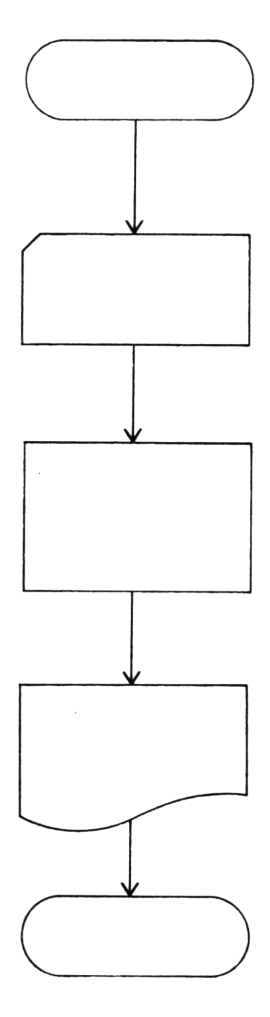
1. Llegar al Instituto.
2. Si está el compañero-a ir a 3. Si no, ir a 4.
3. Hablarle de la última conquista e ir a 4.
4. Esperar la llegada del profesor.
5. Entrar con ánimo a clase.

62. Hacer el organigrama del algoritmo de DAR TRES NUMEROS DISTINTOS Y AVERIGUAR SI PUEDEN SER LAS MEDIDAS DE LOS LADOS DE UN TRIANGULO, con las siguientes acciones:

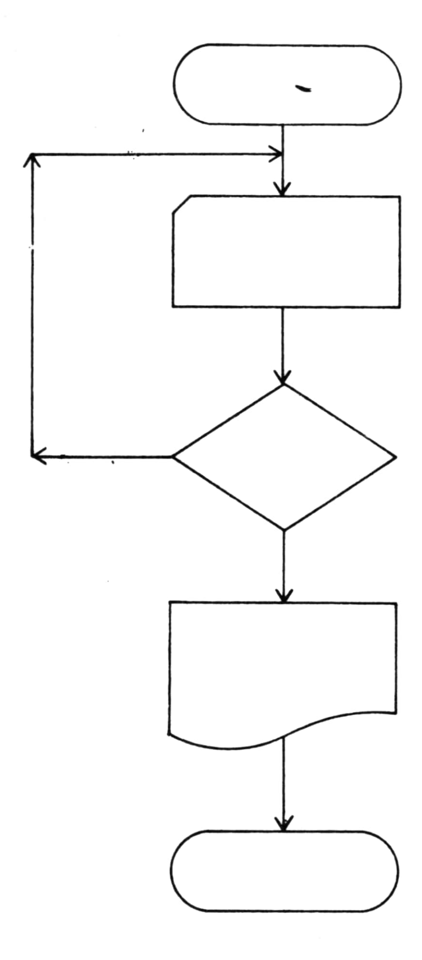
1. Entrar los tres lados.
2. Sumar los dos menores.
3. Si el mayor es menor que esta suma imprimir: "si". En caso contrario imprimir "no".

*63. Hacer el organigrama de un alegoritmo correspondiente al siguiente problema: CALCULAR EL IMPUESTO A PAGAR POR UNA DETERMINADA CANTIDAD C BAJO EL SIGUIENTE CRITERIO: EL 10% SI C ES MENOR O IGUAL QUE 15000, EL 20% SI C ESTA ENTRE 15000 Y 45000 Y EL 30% SI C ES MAYOR O IGUAL QUE 45000.

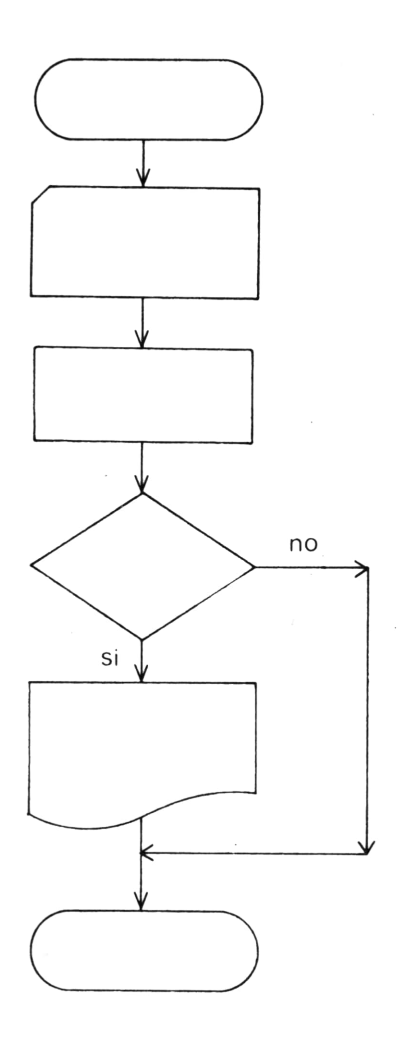
64. Completar el siguiente organigrama para un problema concreto.



65. Completar el siguiente organigrama para un problema concreto.

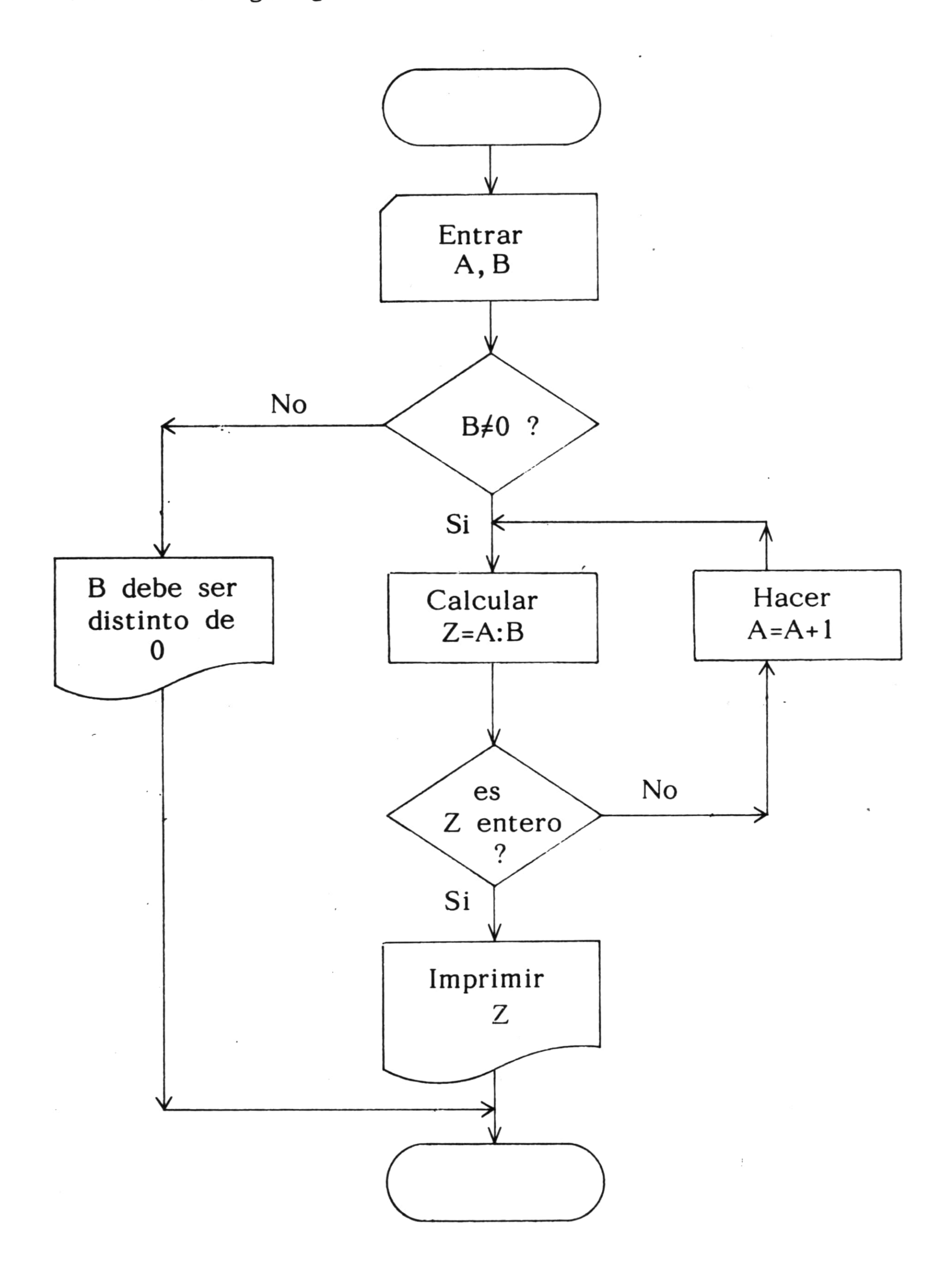


66. Completar de dos maneras distintas el diagrama de flujo siguiente de manera que representen a dos problemas distintos:



*67. Hacer un organigrama en el que se reflejen las siguientes acciones: Entrar dos números y efectuar su diferencia (primero menos segundo). Si esta diferencia es positiva o nula, efectuar la raiz cuadrada e imprimirla. Si es negativa imprimir el mensaje "No se puede realizar la raiz cuadrada de un número negativo".

68. En el organigrama adjunto:



a) Explicar qué problema resuelve.
b) ¿Cómo se expresa con funciones matemáticas del BASIC la acción averiguar si z es entero?

69. Hacer el diagrama de flujo para el cálculo de n!(siendo n! = 1.2.3.4.....n).

70. Hacer el diagrama de flujo para el siguiente problema: Dados n números $A_1, A_2, \ldots, A_n$, imprimir el menor de ellos.

71. Explicar a qué se llama programa fuente y programa objeto.

72. Explicar qué diferencia hay entre un compilador y un intérprete.

73. ¿Por qué es conveniente numerar las líneas de instrucción de 10 en 10?

74. ¿Cuáles de las siguientes expresiones se pueden utilizar como variables?
a) X$, b) 1$, c) $X , d) $2 , e) AB$

75. ¿Qué aparecerá en la pantalla al ejecutar el siguiente programa?

```
10 LET A$ = "Manolo"
20 LET B$ = "Lola"
30 LET C$ = "Lucas"
40 PRINT A$;" enreda a "; B$;" y ";C$
50 END
```

76. ¿Qué aparecerá en la pantalla al ejecutar el siguiente programa?

```
10 LET A = 5 : LET B = 4 : LET C$ ="Antonio" :
       LET D$ = "Luisa"
20 PRINT "5 por 4 es igual a ";A*B:PRINT C$;
       " y ";D$;" estudian el mismo curso"
30 END
```

*77. ¿Qué aparecerá en la pantalla al ejecutar el siguiente programa?

```
10 LET LADO = 5
20 PRINT "El area del cuadrado de lado 5 es igual
       a ";LADO ↑ 2
30 END
```

78. Si ejecutamos el programa:

```
10 LET A$ = "NUMERADOR"
20 LET B$ = "DENOMINADOR"
30 PRINT TAB(17);A$
40 PRINT TAB(16);"____________"
50 PRINT TAB(16);B$
60 END
```

¿qué imprimirá?

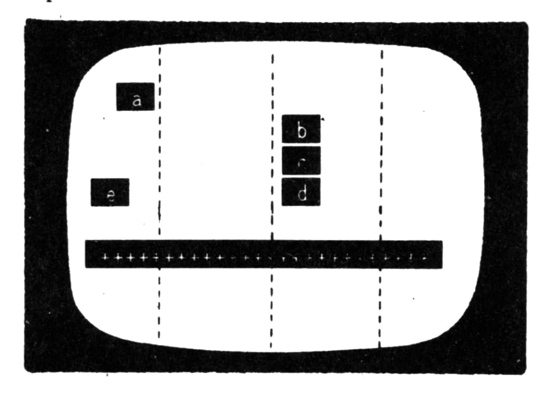
79. ¿Qué imprimirá la ejecución del programa siguiente?

```
10 LET A$ = "Diez"
20 LET B$ = "mil"
30 PRINT A$ + B$
40 END
```

* 80. ¿Hay algún error en el siguiente programa?

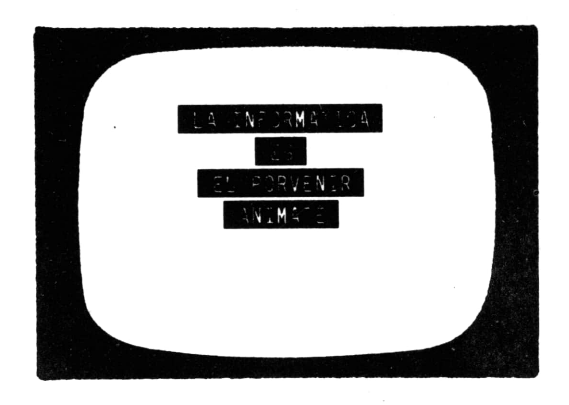
```
10 LET A$ = "6"
20 LET B$ = "5"
30 PRINT A$ + B$
40 PRINT A$ * B$
50 END
```

81. Escribir un programa en el que aparezca el gráfico adjunto en la pantalla:



* 82. Escribir un programa que imprima 3 cadenas ya definidas A$,B$,C$ separadas por una línea en blanco.

83. Escribir un programa para que aparezca centrado en la pantalla el siguiente mensaje:



84. ¿Qué dificultad puede encontrar el ordenador cuando se ejecute la línea

```
50 LET X = (Y - Z) ↑ 0.6
```

en un programa?

85. Si hay una instrucción que es

```
LET X = -A ↑ 3
```

¿Qué valor tomará X si A=3?

86. Si hay una instrucción que es

```
LET X = A ↑ 3
```

¿Qué valor tomará X si A=-3?

87. ¿Cuál debería ser el valor de X y el de Y para que la ejecución del siguiente programa

```
10 LET A = X
20 LET B = Y
30 PRINT SQR(A)
40 PRINT SQR(B)
50 END
```

diese como resultado:

```
2E+9
1E-8  ?
```

88. Escribir en forma de sentencia LET :

$$a)\ X = \frac{A + B}{C.D}\ ;\ b)\ Y = A^3.B\ ;\ c)\ Z = \frac{A^4}{B^2}$$

89. Escribir en forma de sentencia LET :

$$a)\ X = A - \frac{B}{C}\ ;\ b)\ Y = \frac{A}{B\ .\ C}\ ;\ c)\ Z = \sqrt{A^2+B^2}$$

90. Hacer un programa cuya ejecución imprima:

Voy a calcular los siguientes valores:

```
5*6↑(1/5) =
SIN(7*π/5) =
LN(7↑3) =
```

(dando las soluciones de estas expresiones)

* 91. Hacer un programa cuya ejecución imprima el valor de la expresión

$$\sqrt[5]{(531,2.A)^2.\left|tag\ \frac{2\pi B}{3}\right|}$$

para: a) A=3 , B=4,5 ; b) A=7,3 , B=59,8

92. Supongamos que en el ordenador tenemos introducido el siguiente programa:

```
10 LET X = 5
20 LET Y = 10
30 PRINT X + Y
40 LET A = 40
50 LET B = 20
60 PRINT A * B
70 END
```

y sólo queremos que ejecute a partir de la línea 40. ¿Qué debemos pulsar?

93. Si tenemos en el ordenador ya introducido el programa:

```
10 LET X = 10
20 LET Y = 20
30 PRINT X + Y
40 PRINT X * Y
50 END
```

¿qué resultado nos daría en pantalla al pulsar RUN 30?

94. ¿Qué daría la ejecución del siguiente programa?

```
10 LET X = -10
20 PRINT X↑2
30 PRINT SQR(X)
40 END
```

Razonar la respuesta.

95. ¿Cuál es el resultado de la ejecución del siguiente programa?:

```
10 LET X = 5
20 LET Y = 10
```

```
30 PRINT Y/X
40 LET X = 2*X
50 PRINT Y/X
60 LET Y = Y/2
70 PRINT X * Y
80 END
```

*96. Determinar qué imprimirá la ejecución del siguiente programa:

```
10 LET A = 2
20 LET B = -3 : LET C = 4
30 LET D = 5
40 LET X = A + B + 2 * C - D
50 LET Y = X * (A - B)
60 LET Z = Y * D - (X + A)
70 LET Y = Z + 3 * Y - 1
80 PRINT X,Y,Z
90 END
```

* 97. Si introdujéramos el siguiente programa:

```
10 LET A = 25
20 PRINT "A * 2 ="; A*2
30 PRINT "A/A="; A/A
30
40 PRINT "A + A ="; A+A
50 END
```

en un ordenador, ¿qué imprimiría?

98. ¿Qué aparecerá en la pantalla con la ejecución del siguiente programa?

```
10 LET A$ = "MARIA"
20 PRINT "La raiz cuadrada de 25 es "; SQR(25)
30 PRINT "Antonia es amiga de ";A$
40 END
```

99. ¿Qué imprimirá la ejeción del siguiente programa?

```
10 LET X = 5 : LET Y = 7
20 LET Z = X + Y
30 LET P = X * Y
40 END
```

*100. Si introdujéramos el siguiente programa:

```
30 PRINT X + Y
60 PRINT X - Y
40 PRINT X/Y
10 LET X = 40
20 LET Y = 10
50 PRINT X * Y
60 END
```

en un ordenador, ¿daría mensaje de error? En caso negativo, ¿qué imprimiría?

101. Si introdujéramos el siguiente programa:

```
10 LET A = 20 : LET B = 30
20 PRINT "A * B ="; A*B
30 PRINT "A + B ="; A+B
40 END
```

en un ordenador, ¿qué imprimiría?

102. ¿Hay algún error en el siguiente programa?

```
10 LET A$ = Nombre
20 PRINT A$
30 END
```

103. ¿Hay algún error en el siguiente programa?

```
10 LET B = -6
20 LET C = 2
30 LET 2A = 4
40 PRINT B + C + 2A
50 END
```

104. Todas las sentencias LET siguientes son incorrectas. Decir cual es el error.

```
LET A = 24,6
LET B = 30
LET C+1 = C
```

105. ¿Hay algún error en el siguiente programa?

```
10 LET X = 5
20 LET Y = X * X + 3
30 LET X+4 = Z
40 PRINT X,Y,Z
50 END
```

106. ¿Hay algún error en el siguiente programa?

```
10 LET X = 3
20 LET X = X + 3.5
30 PRINT X↑X
40 LET X * 5 = Y
50 PRINT X,Y
60 END
```

* 107. ¿Hay algún error en el siguiente programa?

```
10 LET A = 6
20 LET B = -2
30 LET A+B = X
```

```
40 PRINT X
50 END
```

108. Si un franco francés vale 18,7 pesetas y un dólar 163,5 pesetas (cambio en julio de 1984), hacer un programa que imprima un cuadro con estas equivalencias y el valor en francos y dólares de 525.000 pesetas.

109. Hacer un programa que dé 5 números enteros al azar entre 1 y 50, y calcule su media aritmética y su media geométrica.

110. La ejecución de un programa debe imprimir 6 variables: A,B,C,D,E,F. Escribir este programa de modo que:

a) Las imprima una en cada línea.
b) Las imprima todas seguidas con una separación de dos espacios entre ellas.
c) Imprima dos en cada línea.

*111. Escribir una parte de un programa que intercambie el valor de dos variables A y B ya definidas.

4
ENTRADA DE DATOS. OTRAS SENTENCIAS

1. Sentencia INPUT (entrada)

Al realizar un programa casi siempre interesa que nos sirva para distintos valores de las variables. Si no queremos escribir otra vez dicho programa debemos introducir los datos durante su realización. Esto lo conseguimos con la sentencia INPUT.

Si escribimos:

```
1Ø LET BASE = 5
2Ø LET ALTURA = 6
3Ø PRINT "BASE x ALTURA =";BASE * ALTURA
4Ø END
```

El resultado en pantalla de este programa es:

```
BASE x ALTURA = 3Ø
```

Si queremos que este programa nos sirva para calcular el área de cualquier rectángulo tendremos que poder modificar el valor de las variables.

Los podemos hacer de la siguiente manera:

```
1Ø INPUT B
2Ø INPUT A
3Ø PRINT "BASE x ALTURA ="; B*A
4Ø END
```

Si ejecutamos el programa anterior, debido a la línea 1Ø el ordenador espera que entremos el valor de la base B.

Si por ejemplo pulsamos 6 (y RETURN ó ENTER,...), pasará a la línea 2Ø y esperará a que entremos el valor de la altura A.

Si pulsamos 8 (RETURN ó ENTER,...) continuará la ejecución del programa e imprimirá

BASE x ALTURA = 48



La sentencia INPUT admite también mensajes que deben separarse de la variable por un punto y coma (;) y que se utilizan para indicar al usuario las variables que debe introducir, por ejemplo:

```
1Ø INPUT "BASE =";B
2Ø INPUT "ALTURA =";A
3Ø PRINT "BASE x ALTURA =";B*A
4Ø END
```

Con lo cual el ordenador presenta en pantalla:

BASE =?

(en algunos no aparece el interrogante) y espera que le demos un valor.

Si le damos por ejemplo el 6 y pulsamos RETURN aparece:

ALTURA =?

y vuelve a esperar otro valor.

Si pulsamos por ejemplo el 8 y luego RETURN, aparece en pantalla:

BASE x ALTURA = 48

Para volver a utilizarlo nos basta pulsar RUN y vuelve a aparecer en pantalla:

BASE =?

y podemos introducir datos distintos.

Generalmente la sentencia INPUT admite mas de una variable. Podríamos escribir el programa anterior de la siguiente forma:

```
10 INPUT B,A
20 PRINT "BASE = ";B,  "ALTURA = "; A
30 PRINT "AREA = "; B*A
40 END
```

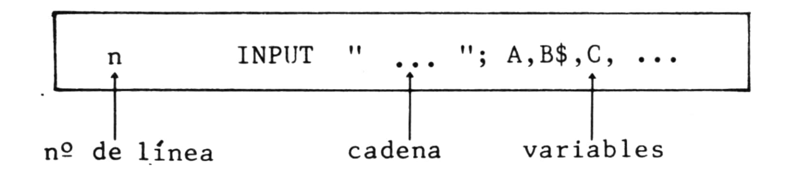
También se utiliza la sentencia INPUT para introducir cadenas, por ejemplo la ejecución de la línea

```
10  INPUT A$
```

esperará a que el usuario introduzca una cadena.

En algunos ordenadores al ejcutarse una línea del tipo A$ da error si la cadena contiene los caracteres , o : Para estos las cadenas de este tipo hay que entrarlas entre comillas.

Las instrucciones que contienen esta sentencia corresponden al siguiente formato:



2. Sentencia NEW (nuevo)

Con la sentencia NEW se borra el programa almacenado en la memoria RAM, asi como todas las variables. (Equivale a desconectar el ordenador y volverlo a conectar).



Se utiliza normalmente como comando.

Pulsar NEW por error destruye por tanto todo el programa almacenado. Si este error se comete después de teclear un largo programa resulta que se pierde todo el tiempo empleado.

3. Sentencia CLS (borrar pantalla)

Con la sentencia CLS se borra el contenido de la pantalla, de manera que el siguiente PRINT empezará a imprimirse en la primera línea.

También puede utilizarse como comando.



Ejemplo

La ejecución del programa

```
1Ø PRINT "Prueba del CLS"
2Ø LET A = 2
```

```
30 PRINT 3 * A
40 PRINT 3 * 5
50 PRINT SQR(25)
60 END
```

imprime:

```
Prueba del CLS
6
15
5
```

Pulsando a continuación el comando CLS queda borrada la pantalla pero no el programa ni las variables a diferencia de la sentencia NEW.

Si se usa como sentencia en una línea de programa borrará la pantalla y continuará la ejecución del mismo.

4. Sentencia LIST (listar)

La sentencia LIST sirve para visualizar la lista de instrucciones del programa almacenado en la memoria RAM, es decir, pulsando LIST (seguido de RETURN, ENTER,...), aparece el listado del programa.

Por ejemplo si tenemos almacenado el último programa de la pregunta nº 1 de este tema, al pulsar LIST (seguido de RETURN,...) aparece en pantalla:

```
10 INPUT B,A
20 PRINT "BASE =";B , "ALTURA =";A
30 PRINT "AREA ="; B*A
40 END
```

Si interesa el listado de la línea n basta pulsar LIST n, por ejemplo en el programa anterior

pulsando LIST 3∅ aparecería en pantalla:

```
30 PRINT "AREA ="; B*A
```

Esta sentencia se utiliza principalmente como comando.

5. Sentencia REM (comentar)

La sentencia REM sirve para introducir comentarios en el programa.

El ordenador ignora esta instrucción salvo para el listado del programa, es decir, si nosotros pulsamos

```
10 REM ESTE PROGRAMA SIRVE PARA CALCULAR POTENCIAS
```

cuando el ordenador efectúe el programa pasa de largo esta línea.

Entonces podemos preguntarnos, ¿para qué sirve esta sentencia? Sencillamente, para que sepamos de qué trata un programa (poniéndole título) o para recordar el por qué damos unas instrucciones determinadas al ordenador. Por ejemplo, si efectuamos un programa para resolver la ecuación de 2º grado podríamos poner:

```
10 REM PROGRAMA PARA RESOLVER LA ECUACION DE
   SEGUNDO GRADO
```

y dentro de las instrucciones poner otra que diga:

```
.
.
.
30 REM CALCULO DEL DISCRIMINANTE
.
```

La única misión que tiene es pues recordar al programador lo que ha hecho y avisar al que lea el programa porqué lo hace.

6. Sentencia STOP (parar)

Hemos visto que con la sentencia END finalizaba el programa. Pero en ocasiones nos interesará parar la ejecución del programa en alguna de sus líneas sin que éste haya acabado. Para ello utilizaremos la sentencia STOP que detiene la ejecución del programa.

En un programa podemos utilizar la sentencia STOP tantas veces como nos haga falta.

7. Comando CONT (continuar)

El comando CONT se utiliza para continuar la ejecución de un programa detenido por la sentencia STOP.

Por ejemplo la ejecución del siguiente programa:

```
10 INPUT A
20 LET X = A * A + 2
30 PRINT X
40 STOP
50 PRINT X + 3
60 END
```

se parará en la línea 40. Si luego pulsamos CONT seguirá la ejecución a partir de la línea 50.

EJERCICIOS RESUELTOS

IV.1. *¿Hay algún error en los siguientes programas?*

a)

```
10 PRINT PROGRAMA DE VOLUMENES DE ESFERAS
20 INPUT R
30 LET V = 4 * π * R ↑ 3 / 3
40 PRINT R,V
50 END
```

b)

```
10 INPUT A
20 LET A = A * 1,65
30 PRINT A
40 END
```

Solución

a) Hay un error en la línea 10. Tendría que ser

```
10 PRINT "PROGRAMA DE VOLUMENES DE ESFERAS"
```

Faltan las comillas.

b) Hay un error en la línea 20. Debería ser

```
20 LET A = A * 1.65
```

La coma debe ser sustituida por un punto.

* * *

IV.2. *¿Qué imprimirá la ejecución del siguiente programa?*

```
10 INPUT A,B
20 REM A ↑ 2 + B ↑ 2
30 PRINT "REM"
40 END
```

Solución

Este programa sólo escribe REM

* * *

IV.3. *¿Qué imprimirá la ejecución del siguiente programa?*

```
10 LET A = 50
20 LET B = 60
30 NEW
40 PRINT A*B
50 END
```

Solución

No imprimirá nada pues la línea 30 borra el programa y por tanto la línea 40 no llega a ejecutarse.

* * *

IV.4. *Escribir un programa en el que, dados los catetos de un triángulo rectángulo cualquiera, nos dé como resultado la hipotenusa y la medida de los angulos agudos del mismo expresados en radianes.*

Solución

Un programa es:

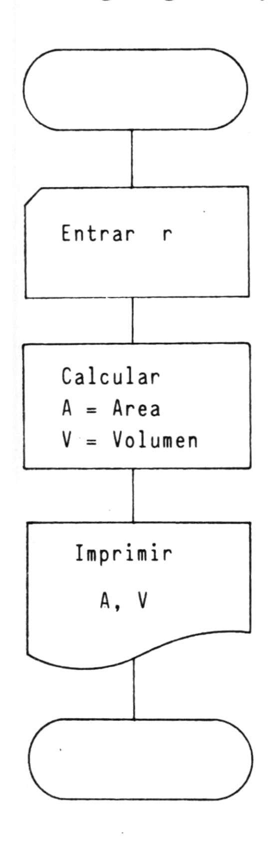
```
10 INPUT "Cateto b="; b
20 INPUT "Cateto c="; c
30 PRINT "Hipotenusa= "; SQR(b ↑ 2 + c ↑ 2)
40 PRINT "B= ";ATN(b/c);" radianes"
50 PRINT "C= ";ATN(c/b);" radianes"
60 END
```

* * *

IV.5. *Escribir un programa que imprima el área y el volumen de una esfera preguntando el radio de esta.*

Solución

El organigrama y el programa correspondiente son:




```
10 REM AREA Y VOLUMEN DE UNA ESFERA
20 INPUT "Radio =";r
30 LET A = 4 * π * r ↑ 2
40 LET V = 4/3 * π * r ↑ 3
50 PRINT "Area = ";A
60 PRINT "Volumen = ";V
70 END
```

* * *

IV.6. *Escribir un programa que transforme en radianes las medidas de los ángulos dadas en grados minutos y segundos sexagesimales. Aplicarlo a los angulos:*

a) 175°; b) 18° 19': c) 59° 13' 15"

Solución

La relación que hay entre grados sexagesimales y radianes es:

$$\frac{x^o}{180^o} = \frac{y\ rad}{\pi\ rad} \Rightarrow y = \frac{\pi x}{180}$$

Un programa es:

```
10 REM PASO DE GRADOS A RADIANES
20 INPUT "Grados =";g
30 INPUT "Minutos =";m
40 INPUT "Segundos =";s
50 LET r = (g + m/60 + s/3600) * π/180
60 PRINT  g;" grados ";m;" minutos " ;
          s ;" segundos son : ";r;" radianes "
70 END
```

Entrando los valores dados en el enunciado da:

a) 175 grados 0 minutos 0 segundos son: 3.0543262 radianes

b) 18 grados 19 minutos 0 segundos son: 0.31968614 radianes

c) 59 grados 13 minutos 15 segundos son: 1.0335985 radianes

* * *

IV.7. *Escribir un programa que calcule el interés compuesto de un capital C preguntando cuál es el capital, el rédito y el tiempo en años.*

$$(C=c(1+r)^t)$$

Solución

El programa es:

```
10 REM INTERES COMPUESTO
20 INPUT "Capital inicial =";c
30 INPUT "Rédito en % =";r
40 INPUT "Tiempo=";t
50 PRINT "El capital final es C = ";c*(1+r/100)↑t
60 END
```

* * *

IV.8. *Escribir un programa que nos dé el valor numérico del polinomio* $ax^4+bx^3+cx^2+dx+e$ *para cualquier valor de* x *y de los coeficientes.*

Solución

Recordemos que el ordenador sólo admite el cálculo de potencias de base positiva o nula. Por ejemplo x↑4 sólo se calcularía para $x \geqslant 0$.

Para obtener valores numéricos para cualquier valor de x daremos dos programas que no usan ↑

```
10 INPUT a,b,c,d,e
20 INPUT x
30 PRINT "Para el polinomio P(x) = ";a;"x↑4 +";
         b;"x↑3 +";c;"x↑2 +",d;"x +";e
40 PRINT "P(";x;") = ";a*x*x*x*x + b*x*x*x +
         c*x*x + d*x + e
50 END
```

El otro programa es el mismo que el anterior cambiando la línea 40 por

```
40 PRINT "P(";x;") = "; x*(x*(x*(a*x+b)+c)+d)+e
```

* * *

IV.9. *El área de un triángulo es:*

$$A = \sqrt{p(p-a)(p-b)(p-c)}$$

donde p es el semiperímetro y a,b,c son las medidas de los lados.

Escribir un programa cuya ejecución dé el área de un triángulo introduciendo a,b,c

Solución

El programa es:

```
10 REM AREA DE UN TRIANGULO
20 INPUT a,b,c
30 LET p=(a+b+c)/2
40 PRINT "Area = ";(p*(p-a)*(p-b)*(p-c))↑0.5
50 END
```

* * *

IV.10. *Escribir un programa cuya ejecución imprima la velocidad con que llegará al suelo un objeto que cae desde una altura h introduciendo h*

$$(v = \sqrt{2gh}\ ,\quad g = 9{,}81\ m/s^2)$$

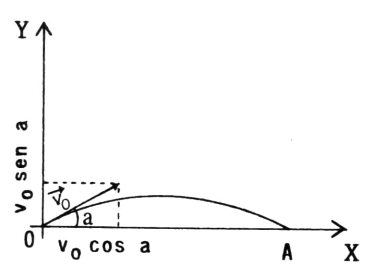
Solución

El programa es:

```
10 REM VELOCIDAD DE CAIDA
20 INPUT "Altura =";h
30 LET V = SQR(2*9.81*h)
40 PRINT "Velocidad = ";v
50 END
```

* * *

IV.11. *En el tiro parabólico indicado en el gráfico, si se considera sin rozamientos, el alcance $\overline{OA}$ viene dado por la expresión:*



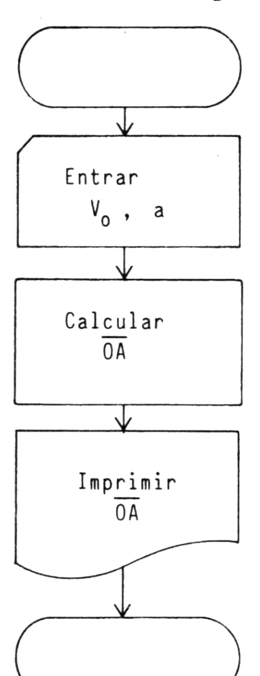
$$\overline{OA} = \frac{V_o^2 \text{ sen}(2a)}{g}$$

siendo g *la gravedad* $g=9{,}81 m/s^2$.

Escribir un programa cuya ejecución dé el alcance del proyectil para cualquier velocidad inicial V_o *y para cualquier angulo* a *en grados.*

Solución

El organigrama y programa correspondientes son:




```
1Ø REM ALCANCE DEL PROYECTIL
2Ø INPUT "Velocidad inicial =";Vo
3Ø INPUT "Angulo de tiro en grados=";a
4Ø LET Y = Vo*Vo*SIN(2*a* /18Ø)/9.81
5Ø PRINT "El alcance es  "; Y
6Ø END
```

Nótese que en la línea 4Ø hemos pasado los grados a radianes para calcular el seno.

* * *

IV.12. *Escribir un programa cuya ejecución resuelva una ecuación de 2º grado con raices reales.*

Solución

La ecuación de 2º grado es:

$$ax^2 + bx + c = 0$$

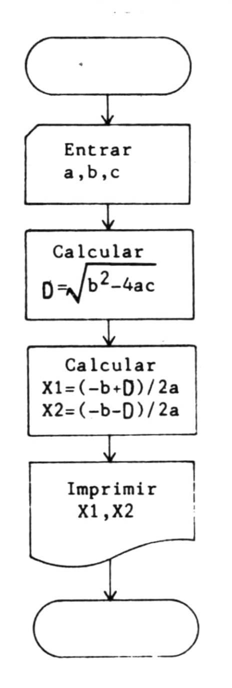
y la solución de la misma es:

$$x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$$

Si $b^2-4ac \geqslant 0$ tiene dos soluciones reales que son:

$$x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a} \qquad x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$$

En el supuesto pues de que $b^2-4ac \geqslant 0$, el diagrama de flujo y el programa correspondiente son:




```
10 REM CALCULO DE LAS RAICES DE UNA
       ECUACION DE SEGUNDO GRADO
20 INPUT "Los coeficientes son:";a,b,c
30 REM CALCULO DEL DISCRIMINANTE
40 LET H = b*b-4*a*c
45 LET D = SQR(H)
50 LET X1 = (-b+D)/(2*a)
60 LET X2 = (-b-D)/(2*a)
70 PRINT "Las raices son :"
80 PRINT "X1 =  ";X1,"X2= ";X2
90 END
```

Para la ecuación

$$x^2-7x+10 = 0$$

una vez introducidos:

1, -7, 1∅

sale en pantalla:

Las raices son: X1=2 X2=5

Observemos que al calcular el discriminante en la línea 4∅, en vez de poner b↑2 hemos puestos b*b. La razón de hacerlo así es que si fuese $b<0$ el ordenador no calcularía la potencia b↑2 y nos daría el mensaje de error.

En caso de no tener raíces reales saldría mensaje de error.

* * *

EJERCICIOS PROPUESTOS

112. ¿Son equivalentes las sentencias LET e INPUT? ¿Hay alguna diferencia entre ellas?

113. ¿Qué ocurrirá si una línea de programa es: n NEW?

114. Si un programa se ha parado debido a una sentencia STOP ¿cómo se puede hacer para que continúe?

115. ¿Puede usarse LIST en una línea de programa?

116. Poner un título adecuado al programa siguiente mediante la sentencia REM

```
20 INPUT R
30 PRINT "L= ";2*π*R
40 END
```

* 117. ¿Qué hará el ordenador al encontrarse con la siguiente línea?

```
n INPUT A, A$
```

118. ¿Qué aparecerá en la pantalla con la ejecución del siguiente programa

```
10 INPUT x
20 LET A = 3 + x
30 LET A = A + 1
40 LET x = x ↑ 2
50 PRINT x - A
60 END
```

si en la línea 1∅ entramos:
a) x=2 ; b) x=3 ?

119. ¿Qué aparecerá en la pantalla con la ejecución del siguiente programa?

```
10 INPUT A,B
20 INPUT C
30 LET X = A*B
40 PRINT C,X
50 NEW
60 PRINT X+3
70 END
```

120. ¿Hace falta modificar alguna línea del siguiente programa para que sea correcto según el título?

```
10 REM Cuadrados de cualquier número
20 INPUT A
30 PRINT A ↑ 2
40 END
```

121. ¿Podría aparecer alguna dificultad en la ejecución del siguiente programa?

```
10 INPUT N
20 PRINT N, SQR(N)
30 END
```

122. Escribir un programa que pregunte el nombre del usuario y luego imprima un saludo a éste

* 123. Escribir un programa en el cual el ordenador pregunte



e imprima las respuestas correspondientes.

124. Supongamos que tenemos un programa con las siguientes líneas:

```
1Ø ...
2Ø ...
25 ...
3Ø ...
4Ø ...
5Ø END
```

y pulsamos LIST 3Ø ¿Qué parte del listado aparecerá en la pantalla?, ¿y si pulsamos LIST 6Ø ?

125. ¿Hay algún error en los siguientes programas?

a)
```
10 INPUT A
20 INPUT B
30 LET Y=A:B
40 PRINT A,B,Y
50 END
```

b)
```
10 INPUT A
20 INPUT B,C
30 LET Y=A+B/A.C
40 PRINT A,B,Y
50 END
```

c)
```
10 INPUT A
20 LET S=A-102.3
30 INPUT B
40 PRINT A,B*B
50 END
```

126. ¿Hay algún error en el siguiente programa?
```
10 INPUT A
20 INPUT B$
30 LET Z = A+B$
40 PRINT Z
50 END
```

127. ¿Hay algún error en el siguiente programa?
```
10 INPUT A
20 INPUT B
30 LET X = (A+B)/C
40 PRINT X
50 END
```
En caso afirmativo señalar cuál. En caso negativo ¿qué hará la ejecución del mismo?

128. Hacer un programa para que calcule el área de una esfera sabiendo su volumen.

129. La fórmula de las anualidades de capitalización es:

$$a = \frac{C \cdot r}{(1+r)\left[(1+r)^t - 1\right]}$$

siendo

$$\begin{cases} C = \text{capital final} \\ r = \text{tanto por uno} \\ a = \text{anualidad} \\ t = \text{tiempo en años} \end{cases}$$

Escribir un programa cuya ejecución nos dé la anualidad a ingresar para formar el capital C al tanto por uno r en t años. Aplicarlo a los casos:

a) C=1300000 ptas., r=0,12, t=8 años
b) C=2000000 ptas., r=0,08, t=5 años

130. La fórmula que nos da las anualidades de amortización es:

$$a = \frac{D \cdot r \cdot (1+r)^t}{(1+r)^t - 1}$$

siendo

$$\begin{cases} D = \text{deuda} \\ r = \text{tanto por uno} \\ t = \text{tiempo en años} \\ a = \text{anualidad} \end{cases}$$

Escribir un programa que nos dé la anualidad a pagar al final de cada año para amortizar una deuda D al tanto por uno r en t años. Aplicarlo a los casos siguientes:

a) D=3000000 ptas., r=0,11, t=10 años
b) D=1200000 ptas., r=0,12, t=5 años

131. Escribir un programa para transformar radianes en grados sexagesimales. Aplicarlo para los ángulos:

a) $\alpha = \pi/5$ rad.; b) $\alpha = \pi/7$ rad.; c) $\alpha = 1,75$ rad.; d) $\alpha = (5/7)\pi$ rad.

(La relación entre radianes y grados sexagesimales es:

$$\frac{\alpha \text{ rad}}{\pi \text{ rad}} = \frac{x^o}{180^o} \quad)$$

132. Modificar el programa anterior para que nos dé en grados, minutos y segundos sexagesimales los ángulos medidos en radianes. ¿Cuál sería el resultado de los cuatro apartados anteriores?

133. Escribir un programa cuya ejecución nos dé la medida de los ángulos agudos de un triángulo rectángulo expresados en grados sexagesimales dados los catetos del mismo.

134. Escribir un programa que convierta un número de 3 cifras dado en una base cualquiera a su expresión en base decimal.

135. Se llama media aritmética de los números: $a_1, a_2, \ldots$ $\ldots, a_n$ al número

$$m_a = \frac{a_1 + a_2 + \ldots + a_n}{n}$$

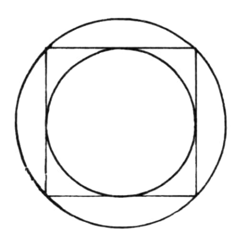
y se llama media geométrica de los mismos al número

$$m_g = \sqrt[n]{a_1 \cdot a_2 \cdot a_3 \cdot \ldots \cdot a_n}$$

Escribir un programa en el que dados 5 números positivos cualesquiera dé como resultado su media aritmética y su media geométrica.

136. Dado un cuadrado de lado a, hacer un programa cuya ejecución nos dé:

a) Perímetro del cuadrado.
b) Area del cuadrado.
c) Perímetro del círculo inscrito.
d) Area del círculo inscrito.
e) Perímetro del círculo circunscrito.
f) Area del círculo circunscrito.
g) Area de la corona circular determinada por ellos,



*137. Dados dos números naturales a y b distintos de cero hacer un programa cuya ejecución nos dé el cociente y el resto de dividir a por b.

138. Si una persona tiene una cuenta con un saldo de 124368 ptas., hacer un programa cuya ejecución pregunte la cantidad que quiere ingresar o sacar e imprima después el nuevo saldo.

139. Escribir un programa que pregunte el precio de un artículo, el tanto por ciento de descuento e imprima la cantidad a pagar.

*140. Escribir un programa que calcule la suma de n términos de una progresión aritmética preguntando el primer término, la diferencia y el número de términos a sumar.

$$(S_n = \frac{a_1+a_n}{2} \cdot n, \quad a_n = a_1+(n-1)d))$$

141. Hacer un programa análogo al anterior para una progresión geométrica de razón positiva y distinta de 1

$$(S_n = \frac{a_n r-a_1}{r-1}, \quad a_n = a_1 r^{n-1})$$

142. Hacer un programa en el que dada una fuerza en Newtons pase a dinas y kilopondios.

(1 Newton = 1/9,81 kilopondios = 10^5 dinas)

143. Siendo la ecuación de un movimiento:

$$e = 2+3t+2,5t^2$$

(t en segundos, e en metros), hacer un programa para hallar el espacio recorrido en un tiempo t. Aplicarlo a los siguientes casos:
a) 2,6 seg.; b) 142 min. 3 seg.; c) 14 min.

144. La velocidad media de un móvil entre los tiempos t_1 y t_2 es

$$v_m = \frac{e_2-e_1}{t_2-t_1} \quad (\frac{\text{espacio recorrido}}{\text{tiempo empleado}})$$

Si la ecuación del movimiento es $e = 2t^2 + (5/2)t + 8$ escribir un programa que nos dé la velocidad media para dos instantes cualesquiera t_1 y t_2.

145. La intensidad de una corriente continua viene dada por I = V/R (amperios), siendo V la diferencia

de potencial en voltios y R la resistencia en ohmios. Escribir un programa que nos dé la intensidad para cualquier número de voltios y de ohmios.

146. Hacer un programa cuya ejecución nos dé la resistencia resultante de conectar 3 resistencias en paralelo. Aplicarlo à los casos:

a) 2 Ω , 3,5 Ω y 7 Ω
b) 1,6 Ω , 5 Ω y 6,2 Ω
c) 5 Ω , 0,6 Ω y 3,2 Ω

$$\left(\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}\right)$$

147. Hacer un programa cuya ejecución nos dé la capacidad resultante de conectar 2 condensadores en serie. Aplicarlo a los siguientes casos:

a) 2 μf y 3,5 μf
b) 2,5 μf y 1,3 μf
c) 6 μf y 2,4 μf

$$\left(\frac{1}{C} = \frac{1}{C_1} + \frac{1}{C_2}\right)$$

*148. La fórmula que nos dá el período T en el movimiento pendular es:

$$T = 2\pi\sqrt{1/g} \quad \text{siendo} \begin{cases} 1 = \text{longitud en cm.} \\ g = 981 \ \text{cm/s}^2 \end{cases}$$

Escribir un programa que nos dé el período para una longitud cualquiera.

149. La relación que hay entre grados centígrados y grados Fahrenheit es:

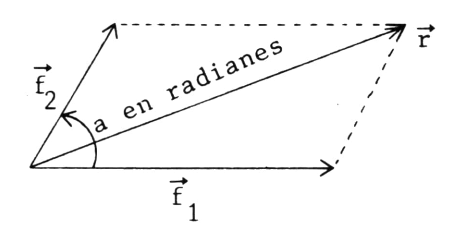
$$\frac{C}{100} = \frac{F-32}{180}$$

Escribir un programa que transforme grados centígrados en grados Fahrenheit y aplicarlo a los casos:
a) C=75º;   b) C=120º

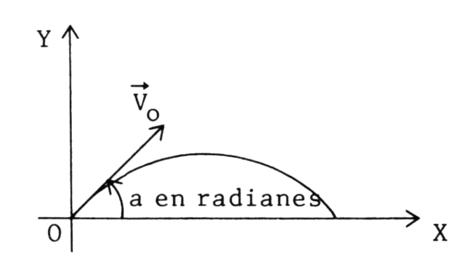
150. Escribir un programa análogo al anterior que transforme grados Fahrenheit en grados centígrados y aplicarlo a los casos siguientes:
a) F=21 ;   b) F = 0

151. Escribir un programa para hallar el módulo de la resultante para unos valores de $|\vec{f}_1|$, $|\vec{f}_2|$ y a según el gráfico adjunto.

$$(|\vec{r}|^2 = |\vec{f}_1|^2+|\vec{f}_2|^2 - 2|\vec{f}_1|\cdot|\vec{f}_2| \cos(\pi-a))$$



*152. Escribir un programa para que en el tiro parabólico representado en el gráfico adjunto:



halle:

a) Las componentes de la velocidad en un instante cualquiera.

b) Las coordenadas del proyectil en este instante.

$$(v = (v_o \cos a,\ v_o \operatorname{sen} a - gt),$$
$$x = (v_o t \cos a,\ v_o t \operatorname{sen} a - \frac{gt^2}{2}),$$
$$g = 9{,}81\ m/s^2)$$

5
SALTOS Y BIFURCACIONES

1. Sentencia GO TO (ir a)

Hasta ahora los programas se ejecutaban en rig roso orden de numeración. Sin embargo en algunos cas es conveniente en una cierta línea saltar a otra ant rior o posterior alterando el orden de numeració En BASIC existe para esto la sentencia GO TO.

Con esta sentencia el ordenador dirige la ejec ción del programa a la línea indicada.



Actúa de la siguiente manera:

Si en un programa aparece

```
.
.
50 GO TO 20
.
.
```

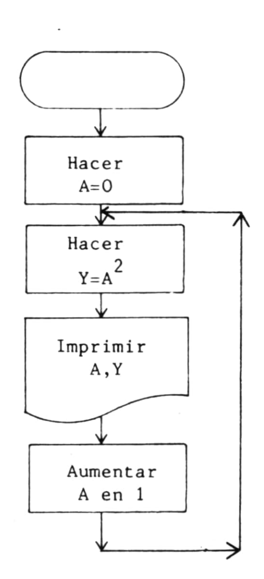
al llegar a la línea 5∅ vuelve a la 2∅ y continúa a partir de la misma.

Ejemplo

Mediante el siguiente programa:

```
10 REM  CUADRADOS DE LOS NUMEROS NATURALES
20 LET  A=0
30 LET  Y=A*A
40 PRINT  A,Y
50 LET  A=A+1
60 GO TO 30
70 END
```

cuyo organigrama es:



aparece en pantalla:

```
0     0
1     1
2     4
3     9
4    16
:     :
```

Nótese que este proceso no acabaría nunca. Se dice que ha entrado en un bucle sin fin. Si queremos parar la ejecución del programa tendremos que pulsar el comando BREAK (romper).

Si después de pulsar BREAK queremos que la ejecución del programa continúe tendremos que pulsar el comando CONT.

Ejemplo

En el último programa de la primera pregunta del Tema IV, si añadimos la sentencia:

```
35 GO TO 1Ø
```

quedando

```
1Ø INPUT  B,A
2Ø PRINT "BASE = ";B , "ALTURA = "; A
3Ø PRINT "AREA = " ; B* A
35 GO TO 1Ø
4Ø END
```

después de imprimir el área de un rectángulo espera que le demos otra vez la BASE Y ALTURA, sin necesidad de pulsar RUN.

Se puede utilizar GO TO como comando de la siguiente manera:

Si pulsamos GO TO n empieza a ejecutarse el programa en la línea n.

GO TO n es equivalente a RUN n con la diferencia de que GO TO n no borra las variables.

2. Operadores de relación

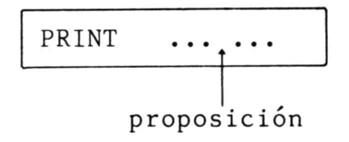
Se llaman así a los operadores que comparan dos números, dos variables (numéricas o alfanuméricas) o dos expresiones.

Son:

=	igual que
<	menor que
>	mayor que
<=	menor o igual que
>=	mayor o igual que
<>	distinto de

Se utilizan mediante PRINT (en forma de sentencia o de comando) o bien con la sentencia IF ... THEN ... que veremos en este Tema.

Escribiendo



dará Ø si la proposición es falsa y 1 si es verdadera. En algunos ordenadores estas respuestas numéricas son distintas, dando Ø si es falsa y -1 si es verdadera.

Ejemplos

a) PRINT 4 = 3 dará como resultado Ø
b) PRINT 4 + 2 = 6 dará como resultado 1
c) PRINT 4 * 3 > 2Ø dará como resultado Ø
d) PRINT 4 ⩽ 4 dará como resultado 1
e) PRINT -3.2 < SQR(5) dará como resultado 1
f) PRINT x + 4 = y + 7 dará Ø ó 1 dependiendo de los valores que tengan las variables x,y

Para comparar cadenas en las que entren los caracteres: espacio, números del Ø al 9 y letras del alfabeto hay que tener en cuenta que el ordenador las considera en el siguiente orden:

espacio < Ø < 1 < 2 < ... < 9 < A < B < C < ... < Z < a < b < c < ... < z

Para comparar las cadenas A$ y B$ lo hace de la siguiente forma:

1) Si el primer carácter de A$ es menor que el primero de B$ ⇒ A$ < B$.

   Si el primer carácter de A$ es mayor que el primero de B$ ⇒ A$ > B$.

2) En caso que ambos primeros caracteres sean iguales pasa a comparar los segundos y así sucesivamente hasta que uno que sea mayor que el otro.

3) Si en este proceso una de las cadenas se acaba antes, ésta es la menor.

Ejemplos

a) PRINT "m" < "n" dará como resultado 1

b) PRINT "a" = "A" dará como resultado Ø
c) PRINT "Mama" < "mama" dará como resultado 1
d) PRINT "32" < "9" dará como resultado 1, puesto que al comparar los dos primeros caracteres el primero es menor que el segundo.
e) PRINT "ALA" = "A LA" dará como resultado Ø, puesto que el segundo carácter de la primera cadena es mayor que el segundo carácter de la segunda cadena.

En el Tema X completaremos la ordenación de las cadenas con todos los caracteres, usando el código ASCII.

3. Operadores lógicos: AND(y) , OR(o) , NOT(no)

Si P y Q son dos relaciones podemos formar las proposiciones siguientes:

1) `P AND Q` que será verdadera cuando lo sean P y Q simultáneamente.

2) `P OR Q` que será verdadera cuando lo sea alguna de las dos.

3) `NOT P` que será verdadera cuando P sea falsa.

Ejemplos

a) PRINT 2 = 3 AND 4 = 2 + 2 dará como resultado Ø, puesto que la primera relación es falsa.

b) PRINT 5 > 1 AND "a" < "he" dará como resultado 1, puesto que las dos relaciones son ciertas.

c) PRINT 4 + 1 = 8 OR 2 * 3 = 6 dará como resultado 1, puesto que la segunda relación es cierta.

d) PRINT NOT (4 = 5) dará como resultado 1, puesto que la relación 4 = 5 es falsa.

<u>**Nota**</u>:

La expresión NOT n, cuando n es un número entero, da -(n+1).

<u>Ejemplo</u>

a) PRINT NOT 3 da como resultado -4
b) PRINT NOT 7 da como resultado -8
c) PRINT NOT -5 da como resultado 4

Al pulsar NOT dos veces equivale a dejarlo como estaba al principio, es decir:

NOT NOT 3 da como resultado 3
NOT NOT -5 da como resultado -5

En algún ordenador la regla dada para los números enteros no es válida.

Resumiendo

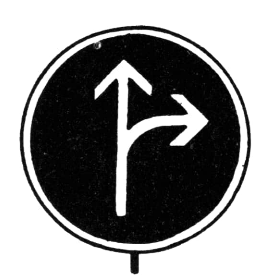
PRINT P AND Q da { Ø si P ó Q son falsas.
                   1 si P y Q son verdaderas.

PRINT P OR Q da { Ø si P y Q son falsas.
                  1 si P ó Q son verdaderas.

PRINT NOT P da { Ø si P es verdadera.
                 1 si P es falsa

PRINT NOT n da -(n+1),
(n entero)

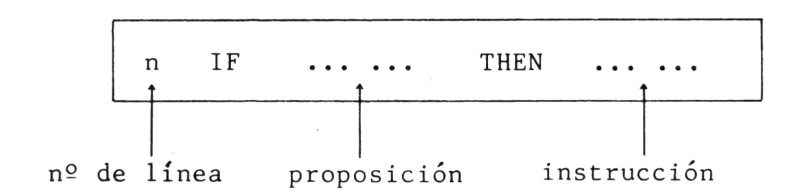
4. Sentencia IF...THEN (si ... entonces)



Los operadores de relación y lógicos se emplean para que el ordenador tome decisiones. Para ello existe la sentencia

IF ... THEN

que se ajusta al siguiente formato:

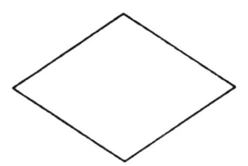


y actúa de la siguiente manera:

Si la proposición es verdadera efectúa la instrucción y si la proposición es falsa la ejecución

del programa pasa a la siguiente línea ignorando el resto de las instrucciones que siguen a THEN.

Esta sentencia corresponde a las bifurcaciones que se han indicado en los organigramas con el símbolo



Ejemplos

a) Si una línea de un programa es:

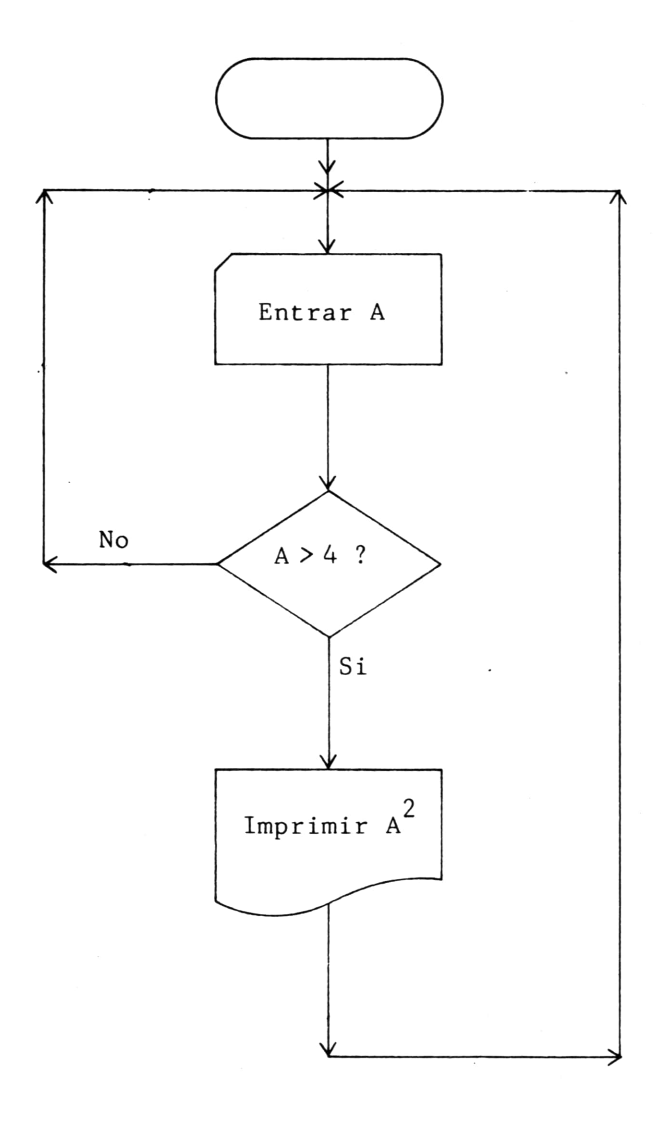
```
IF A>4 THEN PRINT A * A
```

al ejecutarse esta línea se imprimirá el cuadrado de A si A>4, en caso contrario pasará a la línea siguiente

b) El programa:

```
┌→ 1Ø INPUT A
│  2Ø IF A>4 THEN PRINT A*A
└─ 3Ø GO TO 1Ø
   4Ø END
```

que corresponde al organigrama

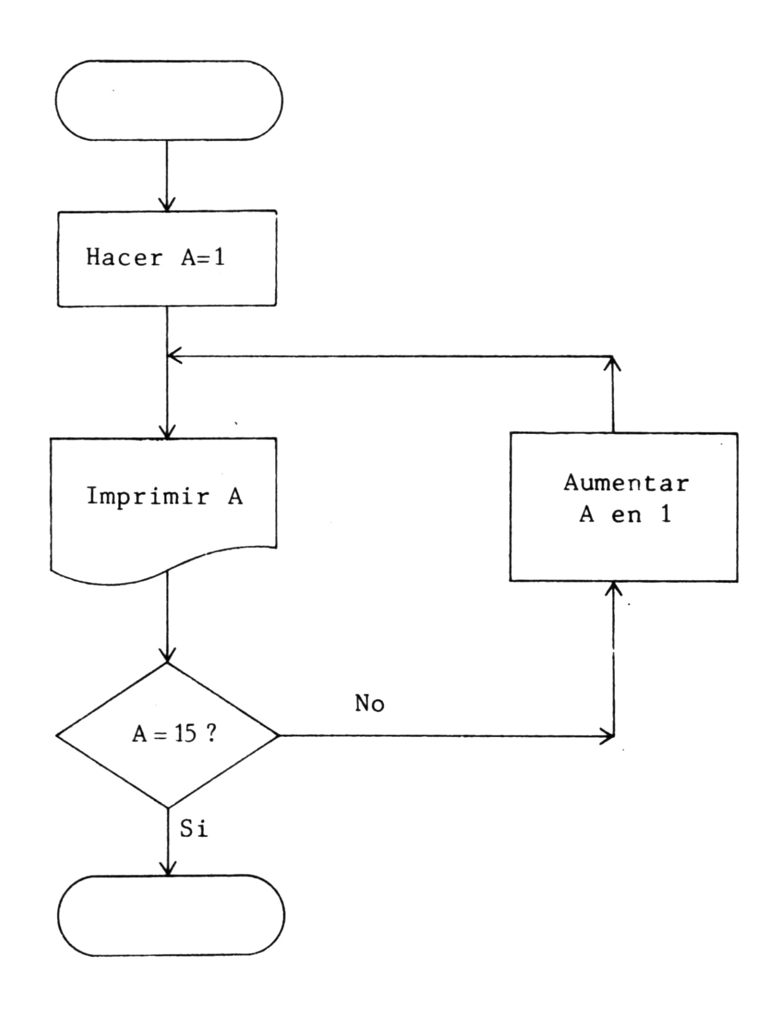


imprimirá el cuadrado de A cada vez que entremos un número $A > 4$, esperando a continuación a que entremos otro valor de A (es un bucle sin fin).

c) El programa:

```
10 REM CONTAR DE 1 A 15
20 LET A = 1
30 PRINT A
40 IF A = 15 THEN GO TO 70
50 LET A = A + 1
60 GO TO 30
70 END
```

correspondiente al organigrama



imprimirá todos los números naturales del 1 al 15.

En algunos ordenadores si la sentencia es:

IF ... THEN GO TO n

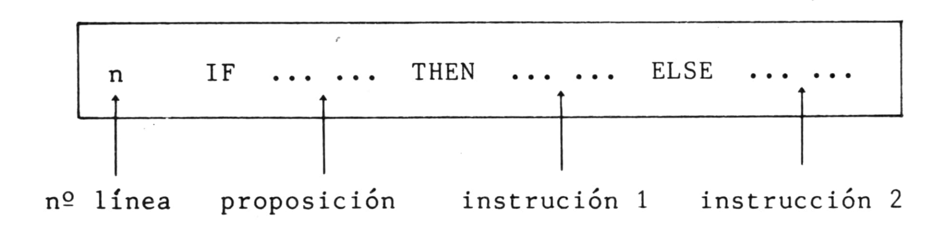
se puede suprimir THEN o bien GO TO; así admiten

IF ... THEN n

o bien

IF ... GO TO n

La sentencia IF ... THEN admite también el formato



que significa:

En caso de ser cierta la proposición se ejecuta la instrucción 1 y si no es cierta se ejecuta la instrucción 2.

5. Dialogando con el ordenador

Estamos ya en condiciones de hacer un programa en el que aparezca un diálogo entre ordenador y usuario.

Estos programas están basados en hacer preguntas e introducir unas variables alfanuméricas (recordar que acaban en $).

Son del tipo:

```
 1Ø  REM DIALOGANDO CON EL ORDENADOR
 2Ø  PRINT "Yo soy un ordenador juguetón. Cómo te
     llamas?": PRINT
 3Ø  INPUT a$
 4Ø  PRINT "Hola " ; a$ ; " Quieres jugar conmigo?
     Contestame sí o no" : PRINT
 5Ø  INPUT b$
 6Ø  IF b$ <> "si" AND b$ <> "no" THEN PRINT "No
     me has entendido bien, tienes que pulsar si o
     bien no" : PRINT : GO TO 5Ø
 7Ø  IF b$ = "no" THEN GO TO 22Ø
 8Ø  CLS
 9Ø  PRINT "Asi me gusta. Jugaremos a adivinar números."
     : PRINT
1ØØ  PRINT "El juego consiste en que yo pensaré un
     número entre 1 y 1000 y tú tendrás que adivinarlo."
     : PRINT
11Ø  PRINT "Para ello te daré pistas diciéndote si
     es menor o mayor. Pero contaré las veces que
     empleas en adivinar el número pensado.": PRINT
12Ø  PRINT "Estoy pensando el número" : LET n = INT(RND*
     1ØØØ)+1
13Ø  PRINT "Pulsa CONT"
14Ø  STOP : CLS
150  PRINT "Ahora ya lo tengo pensado. Empecemos el
     juego. El número que he pensado es el ?": PRINT
16Ø  LET h = 1
17Ø  PRINT "Prueba " ; h
18Ø  INPUT m : PRINT m
19Ø  IF n > m THEN PRINT "Es mayor. Prueba otra
     vez.": LET h = h + 1 : GO TO 17Ø
2ØØ  IF n < m THEN PRINT "Es menor. Prueba otra
     vez": LET h=h+1: GO TO 17Ø.
21Ø  IF n = m THEN PRINT "Has acertado en la prue-
     ba "; h : PRINT : GO TO 23Ø
```

```
22Ø  PRINT "Bien, si no quieres jugar puedes dejar
     el sitio a otro. Ya jugaré contigo otro día"
23Ø  END
```

Obsérvese que al final de algunas líneas está la sentencia PRINT. Sirve para dejar una línea en blanco.

EJERCICIOS RESUELTOS

V.1. *¿Es correcta la línea*

```
100 IF X>80 THEN GO TO "150" ?
```

Solución

No es correcta pues 150 debería ir sin comillas.

* * *

V.2. *¿Qué imprimirá el siguiente programa?*

```
10 LET A = 7
20 GO TO 100
30 LET A = A - 5
40 IF A<5 OR A>20 PRINT A
50 STOP
100 IF A>8 THEN GO TO 30
110 LET A = A + 2
120 GO TO 100
130 END
```

Solución

Este programa imprimirá 4.

* * *

V.3. *¿Es correcto el programa*

```
10 INPUT A,B
20 LET Z = A - B
30 IF Z>0 THEN PRINT A ; "  es mayor que ";B
```

```
40 PRINT B ; "  es mayor que  " ; A
50 END
```

para que dados dos números distintos nos diga cuál es el mayor?

Solución

No es correcto, pues se ejecutará siempre la línea 40, con lo cual en el caso de ser A>B dará siempre como resultado

A es mayor que B
B es mayor que A

* * *

V.4. *Hacer un programa en el cual el ordenador pida un número comprendido entre 1 y 31 ambos incluídos. Si el número introducido está fuera de este margen imprima un aviso y vuelva a pedir el dato de este margen.*

Solución

El programa es:

```
10 INPUT "Número entre 1 y 31" ; A
20 IF A<1 OR A>31 THEN PRINT "número fuera
   del margen" : GO TO 10
30 END
```

* * *

V.5. *Escribir un programa que calcule el impuesto a pagar*

según la cantidad ganada, con los siguientes datos: Debe pagar el 10% si la cantidad es menor o igual que 15000, el 15% si es mayor que 15000 y menor o igual que 45000 y el 20% si es mayor que 45000.

Solución

El programa es:

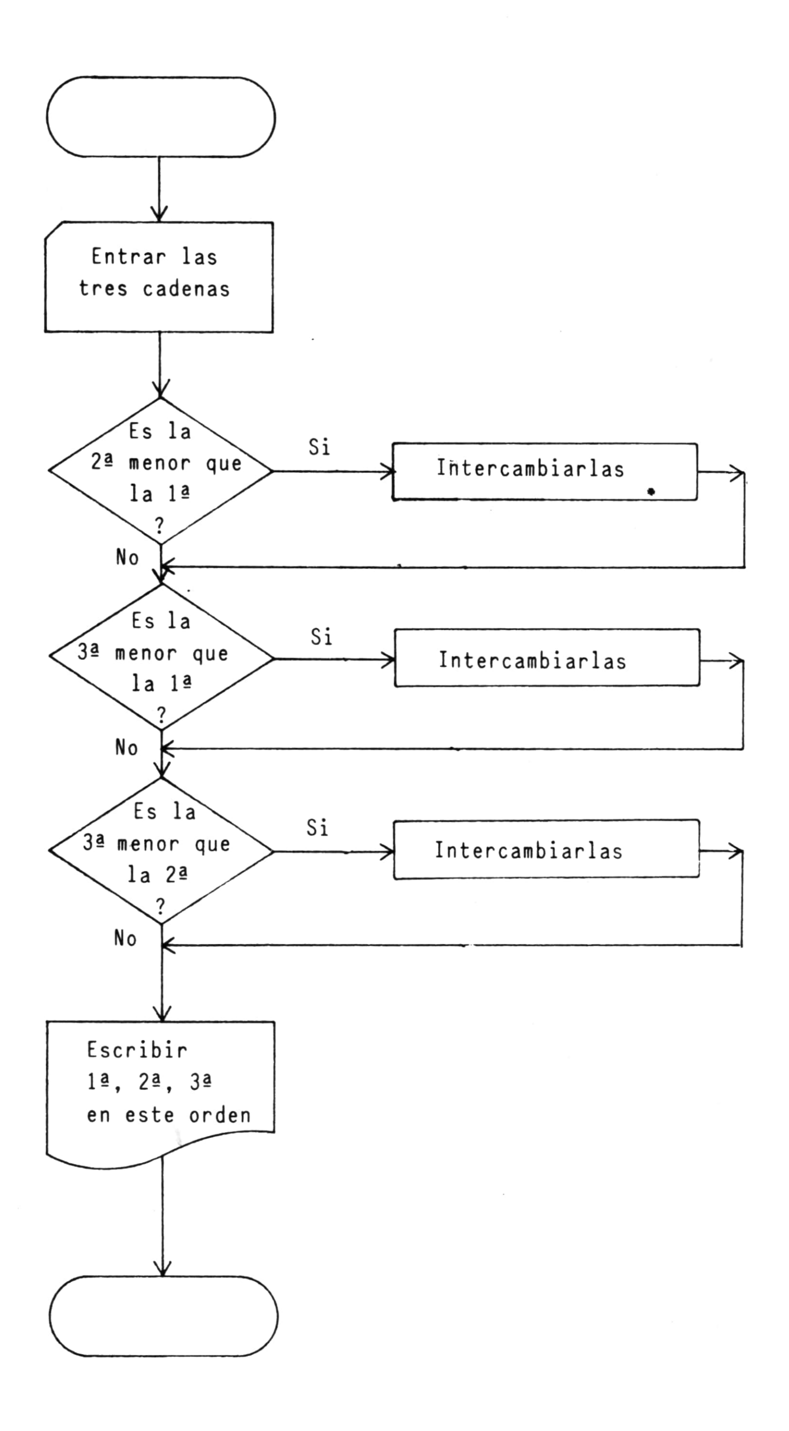
```
10  INPUT "Catidad ganada";C
20  IF C<= 15000 THEN PRINT "I = ";C* 0.1
30  IF C>15000 AND C<= 45000 THEN PRINT "I=";
    C* 0.15
40  IF C > 45000 THEN PRINT "I = "; C* 0.2
50  END
```

* * *

V.6. *Hacer un programa que pida tres cadenas y las escriba en orden alfabético.*

Solución

El organigrama es:



Entrar las tres cadenas
Es la 2ª menor que la 1ª ?
Si
Intercambiarlas
No
Es la 3ª menor que la 1ª ?
Si
Intercambiarlas
No
Es la 3ª menor que la 2ª ?
Si
Intercambiarlas
No
Escribir 1ª, 2ª, 3ª en este orden

El programa es:

```
10 INPUT p$ , s$ , t$
20 IF s$<p$ THEN LET i$ = p$ : LET p$ = s$ :
   LET s$ = i$
30 IF t$<p$ THEN LET i$ = p$ : LET p$ = t$ :
   LET t$ = i$
40 IF t$<s$ THEN LET i$ = s$ : LET s$ = t$ :
   LET t$ = i$
50 PRINT p$
60 PRINT s$
70 PRINT t$
80 END
```

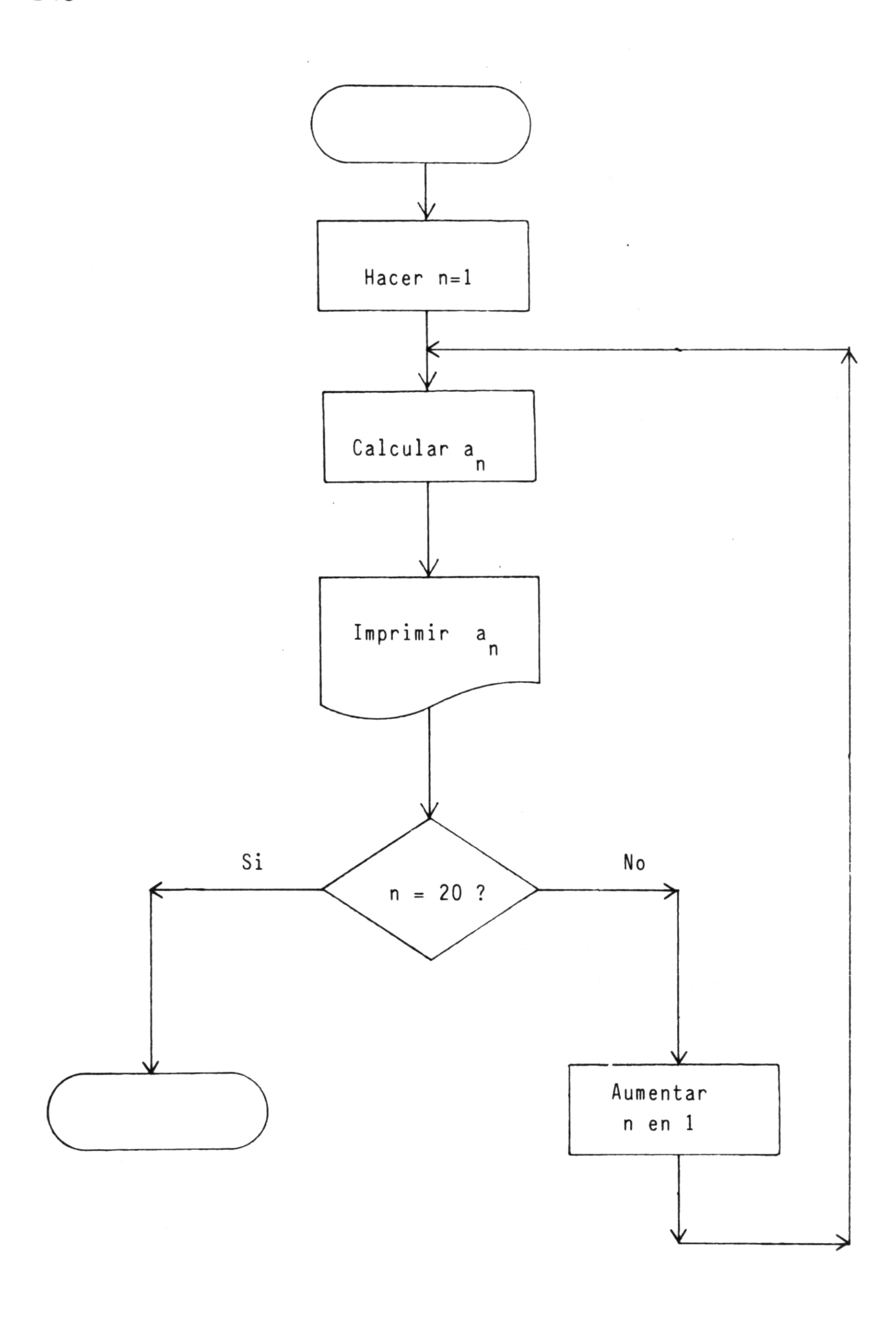
Obsérvese que este proceso es válido para un número mayor de cadenas.

* * *

V.7. *Hacer un programa que escriba los 20 primeros términos de la progresión aritmética* $a_n = 4 + (n-1).5$.

Solución

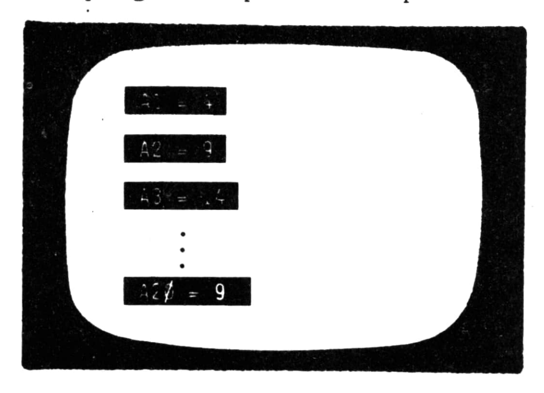
El organigrama y programa correspondientes son:



Hacer n=1
Calcular a_n
Imprimir a_n
n = 20 ?
Si
No
Aumentar n en 1

```
1Ø REM PROGRESION ARITMETICA
2Ø LET N = 1
3Ø LET AN = 4 + (N-1) * 5
4Ø PRINT "A" ; N ; "=" ; AN
5Ø IF N = 2Ø THEN GO TO 8Ø
6Ø LET N = N + 1
7Ø GO TO 3Ø
8Ø END
```

Con este programa aparece en pantalla:

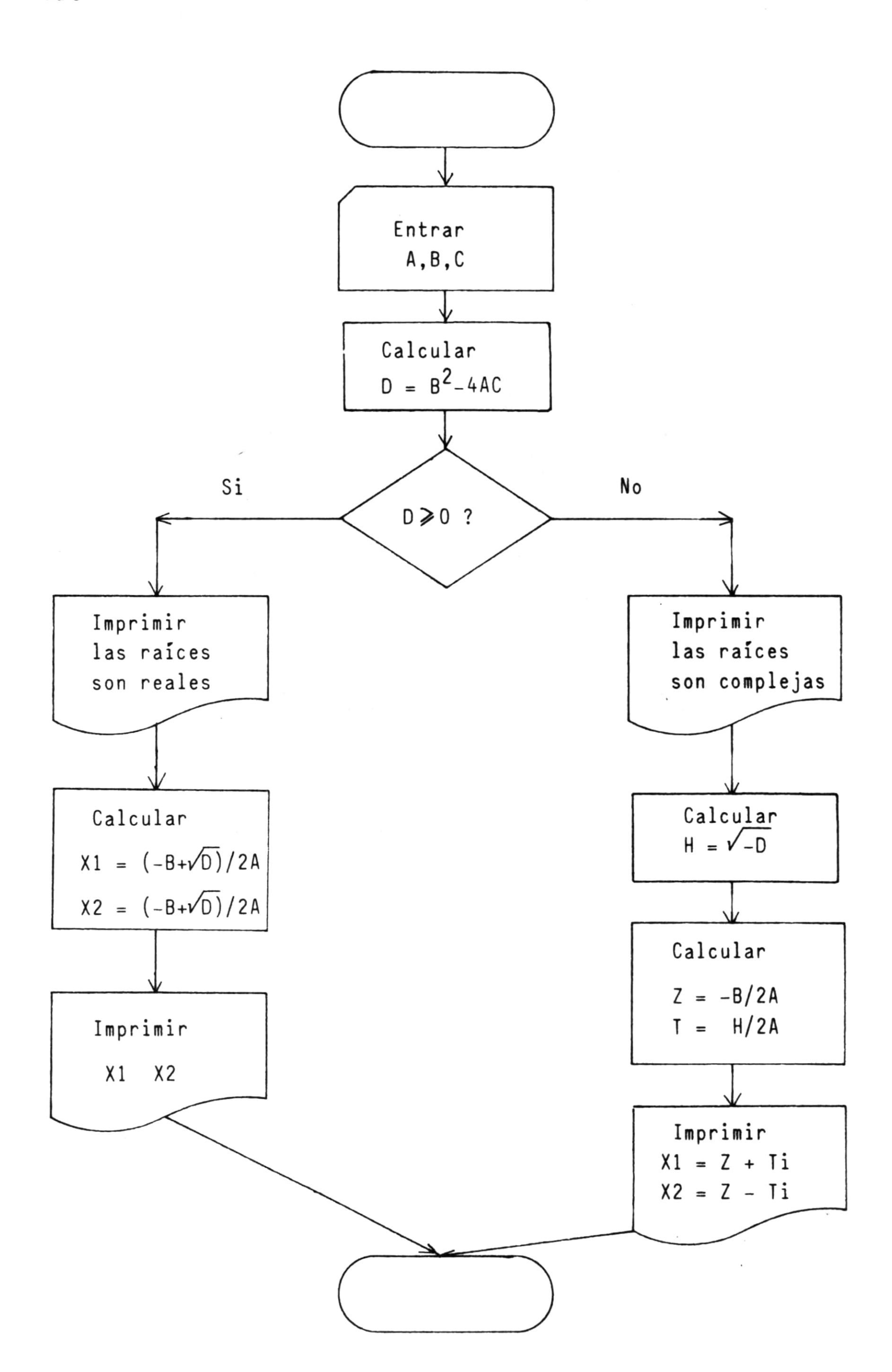


* * *

V.8. *Hacer un programa para resolver una ecuación de 2 grado en el caso general.*

Solución

El organigrama y programa correspondientes son:



Entrar
A,B,C
Calcular
D = B2-4AC
Si
No
D⩾0 ?
Imprimir
las raíces
son reales
Imprimir
las raíces
son complejas
Calcular
X1 = (-B+√D)/2A
X2 = (-B+√D)/2A
Calcular
H = √-D
Calcular
Z = -B/2A
T = H/2A
Imprimir
X1 X2
Imprimir
X1 = Z + Ti
X2 = Z - Ti

```
10 REM RESOLUCION DE LA ECUACION DE SEGUNDO
   GRADO
20 INPUT A,B,C
30 LET D = B *B - 4 *A *C
40 IF D< 0 THEN GO TO 110
50 PRINT "LAS RAICES SON REALES"
60 LET E = SQR(D)
70 LET X1 = (-B+E)/(2 *A)
80 LET X2 = (-B-E)/(2 *A)
90 PRINT "X1=";X1 , "X2=";X2
100 GO TO 160
110 PRINT "LAS RAICES SON COMPLEJAS"
120 LET  H = SQR(-D)
130 LET Z = -B/(2 *A)
140 LET T = H/(2 *A)
150 PRINT "X1=" ; Z ; "+" ; T ; "i" , "X2=" ;
   Z ; "-" ; T ; "i"
160 END
```

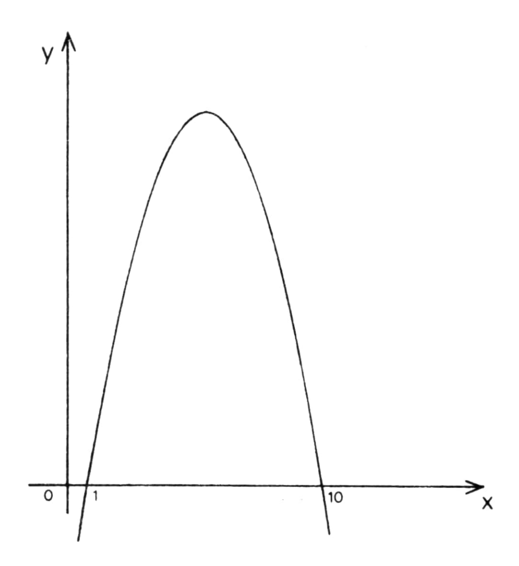
Poniendo A = 1 , B = -7 y C = 10, aparece en pantalla:

```
LAS RAICES SON REALES
X1 = 5            X2 = 2
```

* * *

V.9. *Escribir un programa para obtener el máximo relativo de la función $f(x) = -x^2+11x-10$ con un error menor que una centésima para la x y que se halla en el intervalo $[1,10]$.*

<u>Solución</u>



El método consistirá en hallar f(x) desde x = 1 incrementando la x en 0,01. Se observa la correspondiente f(x) hasta encontrar un valor de f(x) que sea menor que el anterior.

El programa es:

```
10 LET x = 1
20 LET y1 = -x * x + 11 * x - 10
30 LET x = x + 0.01
40 LET y2 = -x * x + 11 * x - 10
50 IF y2 > y1 THEN GO TO 20
60 PRINT "x= " ; x , "f(x)= " ; y1
70 END
```

Nota

Para otra función habrá que modificar el programa.

* * *

V.10. *Escribir un programa que calcule los 100 primeros términos de la sucesión de término general*

$$a_n = \sqrt{n^2+n-1} - \sqrt{n^2+1}$$

imprimiendo n , a_n y la diferencia entre a_n y el límite de la sucesión, sabiendo que dicho límite es 0,5.

Solución

El programa es:

```
10 LET n = 1
20 LET an = (n↑2 + n - 1)↑ 0.5 - (n↑2 + 1)↑
   0.5
30  PRINT n ; TAB(6); "a" ; n ; "=" ; an ;
   TAB(20); an - 0.5
40 IF n = 100 THEN GO TO 70
50 LET n = n+1
60 GO TO 20
70 END
```

* * *

V.11. *La derivada de una función en un punto es:*

$$f'(a) = \lim_{x \to a} \frac{f(x) - f(a)}{x - a}$$

Hacer un programa en el que aparezcan los términos de la sucesión (a + 1/n) que tiende a a y comprobar que

$$sen'(a) = cos(a)$$

Solución

El programa es:

```
10 INPUT a
20 LET n = 1
30 LET x = a + 1/n
40 LET y = (SIN(x) - SIN(a))/ (x - a)
50 PRINT x ; TAB(20); y - COS(a)
60 IF y - COS(a) < 0.0000001 GO TO 90
70 LET n = n + 1
80 GO TO 30
90 END
```

EJERCICIOS PROPUESTOS

153. ¿Es correcta la línea de programa:

```
180 GO TO 230 IF X>0 ?
```

154. ¿Cómo se escribiría en una línea de programa la siguiente instrucción?:

Si $x < y < z$ entonces $h = z - y$

155. ¿Es correcta la siguiente parte de programa:

```
.
.
.
140 IF X>10 THEN 150
150 LET Z = X*Y
.
.
.
```

si se pretende asignar a la variable Z el valor X*Y sólo en el caso de ser X>10 ?

156. ¿Es correcta la instrucción

```
IF X<0 OR Y<0 THEN PRINT "No tiene solución"
```

para que imprima este mensage si uno sólo de los dos X ó Y es negativo?

157. ¿Qué ocurriría si en un programa aparece la siguiente línea?

```
.
.
.
50 GO TO 50
.
.
.
```

158. Una parte de un cierto programa es:

```
.
.
1ØØ GO TO 12Ø
11Ø PRINT "Fijate bien"
12Ø PRINT "Repasa un poco"
.
.
```

Sin saber lo que hay en las líneas anteriores de la 1ØØ, ¿podemos asegurar que nunca se escribirá la expresión "Fijate bien"?

159. ¿Qué daría como resultado la ejecución del siguiente programa?:

```
1Ø LET N = 1
2Ø PRINT N
3Ø LET N = N + 2
4Ø GO TO 2Ø
5Ø END
```

Dibujar el organigrama correspondiente.

160. En el siguiente programa:

```
1Ø INPUT A
2Ø IF A>= 6Ø THEN PRINT "Hola": STOP
3Ø IF A<= 4Ø THEN PRINT "Adios": STOP
4Ø PRINT "Otra vez será"
5Ø END
```

Explicar cuál sería el resultado cuando el número A introducido es: a) menor o igual que 4Ø, b) mayor que 4Ø y menor que 6Ø, c) mayor o igual que 6Ø.

161. Escribir una línea de programa que haga lo siguien-

te: Si una variable A es par y mayor que 4Ø la imprime.

162. Escribir un programa en el que entrando un número natural cualquiera, si es par que dé como respuesta el mismo número aumentado en 1ØØ, mientras que si es impar imprima el triple de dicho número.

163. Hacer un programa que escriba 1ØØØ veces "Arbol se escribe sin H".

164. ¿Para qué sirve el siguiente programa?:

```
1Ø INPUT "Nombre:";N$:INPUT "Primer apellido:";P$
2Ø LET X = Ø : LET M=Ø
3Ø INPUT "Asignatura:";A$:INPUT "Nota:";N
4Ø LET M = M + N
5Ø IF X = 7 THEN GO TO 8Ø
6Ø LET X = X + 1
7Ø GO TO 3Ø
8Ø PRINT "La nota media de  ";N$;" ";P$;
   " es ";M/7
9Ø END
```

165. ¿Qué daría como resultado la ejecución del siguiente programa?:

```
1Ø LET I = 1
2Ø PRINT TAB(I);I
3Ø LET I = I + 1
4Ø IF I< = 2Ø THEN GO TO 2Ø
5Ø END
```

166. ¿Qué imprimiría la ejecución del siguiente pro-

grama?:

```
10 LET N = 1
20 PRINT TAB(N);N
30 LET N = N + 1
40 IF N >10 THEN GO TO 70
50 GO 20
70 LET N = N - 1
80 PRINT TAB(N);N
90 IF N> 0 THEN GO TO 70
100 END
```

167. Escribir un programa que dé como resultado el valor de una compra y el valor a pagar con un descuento según la tabla siguiente:

- Si el valor de la compra es menor que 1000 ptas. el 2%
- Si es mayor o igual que 1000 ptas. y menor que 5000 el 3%
- Si es mayor o igual que 5000 ptas. y menor que 10000 el 5%
- Si es mayor o igual que 10000 ptas. el 8%

*168. Hacer un programa que pida tres números distintos e imprima el menor de ellos.

*169. En una tienda hay 10 clases de artículos: A1,A2,A3, A4,A5,A6,A7,A8,A9,A10. El número correspondiente de unidades de cada uno es: a, b, c, d, e, f, g, h, i, j.

Hacer un programa en el que introduciendo a,b,...,j y el número de unidades vendidas a cada cliente, imprima al final del día el número de unidades que queda de cada clase.

170. Escribir un programa que pida al usuario n números positivos e imprima la suma de todos ellos. (El usuario indicará que se han acabado los datos introduciendo el número -1).

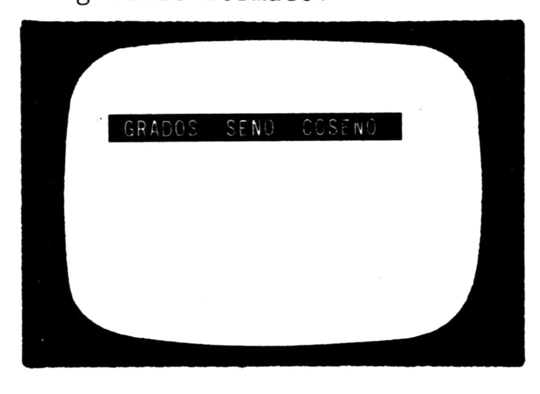
171. Escribir un programa que imprima la letra L con 1Ø asteriscos de altura por 7 de base.

172. Hacer un programa para obtener una tabla de números naturales y sus raices cuadradas.

173. Hacer un programa para obtener una lista de los números naturales y sus cuadrados del 1 al 2Ø.

174. Escribir un programa en el que entrando 3 números distintos cualesquiera, dé como resultado su ordenación de menor a mayor.

*175. Escribir un programa que imprima unas tablas de trigonometria de grado en grado desde 0º a 90º en columnas con el siguiente formato:



(Recordar que el ordenador funciona en radianes)

176. Escribir un programa que imprima los valores de la función seno desde 30º hasta 45º dando los resultados de 30' en 30'.

177. Escribir un programa que imprima una tabla de logaritmos en base 7 del 1 al 1ØØ dejando un espacio cada cinco resultados. ($\log_7 x = LN\ x/LN\ 7$).

*178. Escribir un programa que imprima los términos de la sucesión $a_n = (1+1/n)^n$

179. Escribir un programa que imprima las tablas de multiplicar del 7 y del 9 desde 1 hasta 1Ø.

*180. Las potencias de la unidad imaginaria son: $i^0 = 1$, $i^1 = i$, $i^2 = -1$, $i^3 = -i$. Para $n \geqslant 4$ es $i^n = i^r$ siendo r el resto de dividir n por 4.

Escribir un programa en el que entrando un número natural cualquiera, imprima el resultado de i^n.

181. En una cierta asignatura, la calificación final del curso se calcula así: Nota =(E1 + 2.E2 + 3.E3)/ 6 siendo E1 la nota de la 1ª evaluación. E2 la de la 2ª y E3 la de la 3ª. Escribir un programa que sirva para evaluar esta asignatura.

182. Escribir un programa en el que introduciendo el peso de cada una de las personas de un conjunto, imprima el número de las que pesen entre 65 y 7Ø Kg. (excluyendo los extremos).

183. Escribir un programa que imprima el valor de la suma:

$$1 + 1/2 + 1/3 + 1/4 + \ldots\ldots + 1/312$$

*184. Hacer un programa que genere 1000 números enteros al azar del 1 al 100 (los dos inclusive) e imprima cuantos de ellos están comprendidos entre el 41 y 60 (los dos inclusive). Comprobar si la frecuencia relativa se acerca a la probabilidad que en este caso es 0,2.

185. Hacer un programa en el que introduciendo un número, pruebe si su cuadrado es menor que 2112, y en este caso halle:

a) El radio del círculo que tiene por área el cuadrado de dicho número.

b) La longitud de la circunferencia correspondiente.

En caso contrario, calcule la diferencia que hay entre el cuadrado de dicho número y 2112.

186. Escribir un programa que dé las imágenes de la función

$$y = \sqrt{\frac{x^2+x+1}{x+3}}$$

para valores positivos de x de décima en décima parándose cuando el valor de la función sea mayor que 103,4.

187. Escribir un programa en el que dando el peso (P) de un cuerpo, su volumen (V) y la densidad (D) del fluido en el cual se sumerge, averigüe si flotará o se hundirá.

188. Escribir un programa que dé los 2Ø primeros términos de una progresión geométrica ($a_n = a_1 . r^{n-1}$) y la suma de los mismos. Se entrará el primer término y la razón.

*189. Hacer un programa para calcular el factorial de un número entero positivo (n! = n(n-1)(n-2).....2.1).

190. Escribir un programa en el que entrando un número natural mayor que 1, responda si dicho número es primo o compuesto.

*191. Escribir un programa que halle los factores primos de un número entero positivo mayor que 1.

192. Escribir un programa que calcule el máximo común divisor de dos números a y b enteros positivos, siguiendo el organigrama del algoritmo de Euclides de la pregunta 2 del Tema III.

193. Para calcular el número π, Vieta (1540-1603) utilizaba la sucesión:

$$\frac{2}{\pi} = \frac{\sqrt{2}}{2} \cdot \frac{\sqrt{2+\sqrt{2}}}{2} \cdot \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2} \cdots\cdots$$

Escribir un programa que calcule el número π utilizando dicha sucesión, escribiendo los sucesivos términos de la misma.

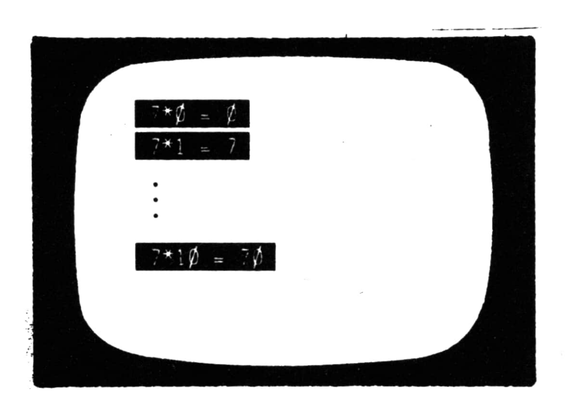
6
BUCLES

1. Sentencia FOR-NEXT

Si queremos confeccionar la tabla de multiplicar del 7 desde 7*Ø hasta 7*1Ø, podemos escribir el siguiente programa:

```
1Ø  LET  X=Ø
2Ø  LET  Y=7
3Ø  PRINT Y;"*";X;"=";Y*X
4Ø  LET  X=X+1
5Ø  IF  X<=1Ø THEN GO TO 3Ø
6Ø  END
```

cuya ejecución da



Obsérvese que este programa consiste en multiplicar 7 por X variando de la X desde 0 hasta 10.

La sentencia FOR-NEXT resuelve problemas de este tipo de forma más directa.

Utilizando esta sentencia el programa anterior podría escribirse de la siguiente manera:

```
10   LET Y=7
20  ┌FOR X=0 TO 10
30  │PRINT Y;"*";X;"=";Y*X
40  └NEXT X
50   END
```

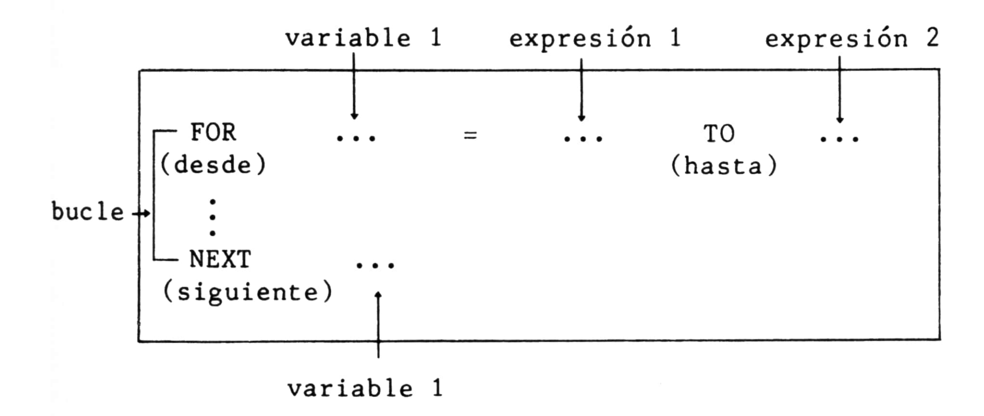
Las líneas 20, 30 y 40 forman un bucle que se inicia para X=0 repitiéndose este hasta X=10 aumentando X de 1 en 1.

Cuando la ejecución del programa pasa por la línea 20 asigna a X el valor 0 y almacena en memoria el valor 10.

La instrucción NEXT X tiene 3 funciones:

1) Aumenta X en 1.
2) Observa si la nueva X es menor o igual que 10.
3) En caso afirmativo envía la ejecución del progra a la línea 30. En caso contrario continúa la ejec ción del programa en la línea siguiente del NEXT.

El formato de esta sentencia es:



En general cuando la ejecución del programa pasa por FOR asigna a la "variable 1" el valor de la "expresión 1" y almacena en memoria el valor de la "expresión 2".

La instrucción NEXT ... tiene 3 funciones:

1) Aumenta la "variable 1" en 1.
2) Observa si la nueva "variable 1" es menor o igual que la "expresión 2".
3) En caso afirmativo envía la ejecución del programa a la línea que sigue al FOR. En caso contrario sigue la ejecución del programa en la línea siguiente del NEXT.

Nota

En algunos ordenadores en el caso de que aparezca un solo bucle en el programa, basta poner NEXT al final del bucle, es decir, se puede omitir el poner la "variable 1". En caso de aparecer varios bucles si no ponemos el nombre de la variable al cerrarlos con NEXT, el ordenador entenderá que debe cerrar en el orden inverso al de la entrada como si se tratase de paréntesis. En otras palabras, en caso de haber más de un bucle, el primer NEXT sin variable cierra el más interior.

Puede haber un bucle dentro de otro, se llaman entonces <u>bucles anidados</u>.

<u>Ejemplos</u>

a)

```
10 ┌FOR A = 1 TO 3
20 │  ┌FOR B = 0 TO 5
30 │  │PRINT "N=" ; 10 * A + B
40 │  └NEXT B
50 │ PRINT
60 └NEXT A
70   END
```

Obsérvense que para cada valor de A, B tiene que tomar los valores desde 0 hasta 5. Así en pantalla tendremos:

```
N = 10
N = 11
N = 12
N = 13
N = 14
N = 15

N = 20
N = 21
N = 22
N = 23
N = 24
N = 25

N = 30
N = 31
N = 32
N = 33
N = 34
N = 35
```

b)

Un programa sencillo para que en pantalla aparezcan todos los capicúas de tres cifras es:

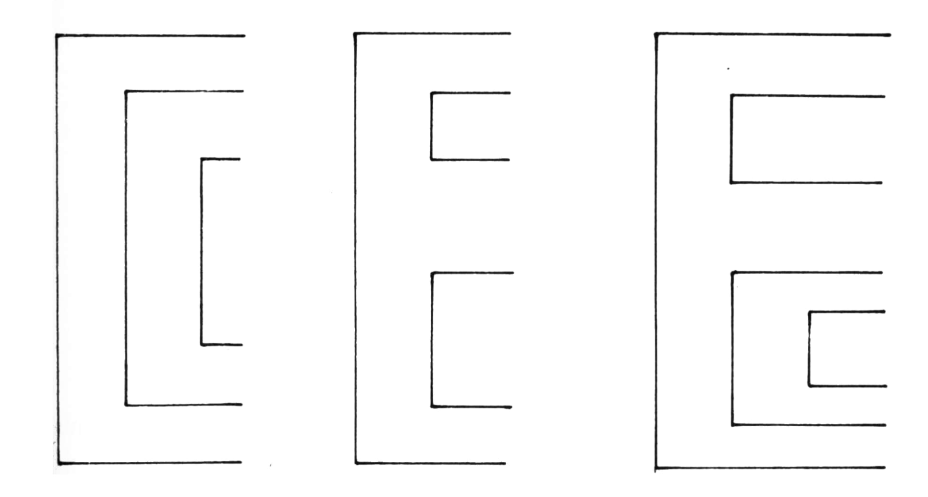
```
1Ø ┌FOR X = 1 TO 9
2Ø │  ┌FOR Y = Ø TO 9
3Ø │  │PRINT 1ØØ  X + 1Ø  Y + X
4Ø │  └NEXT Y
5Ø └NEXT X
6Ø  END
```

Con este programa aparece en pantalla:

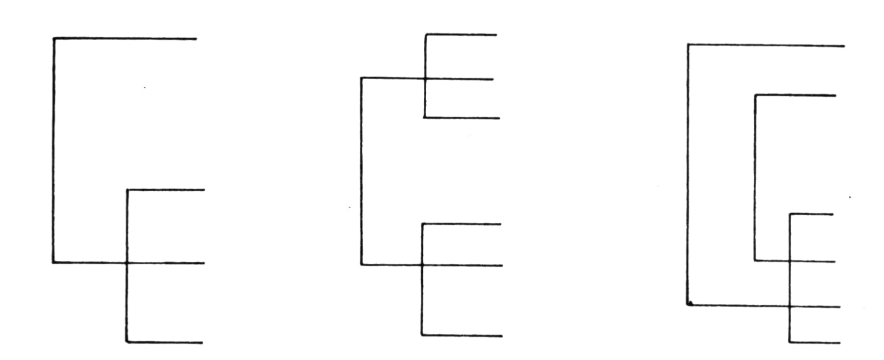
```
1Ø1
111
121
.
.
.
989
999
```

En los ejemplos anteriores aparecen dos bucles "anidados", es decir, que en el mismo programa aparece un bucle dentro de otro bucle. Pueden anidarse más de dos bucles cuando no se interfieren unos con los otros.

Son correctos los siguiente bucles anidados:



No son correctos:



2. STEP(paso): Incremento de la variable

En la sentencia FOR X = N TO M, los distintos valores que va tomando la variable X se incrementan en una unidad. Si queremos modificar este incremento lo podemos hacer de la forma siguiente:

Si escribimos

```
┌FOR X = N TO M STEP H
│.
│.
│.
└NEXT X
```

NEXT X incrementará X en H.

Ejemplo

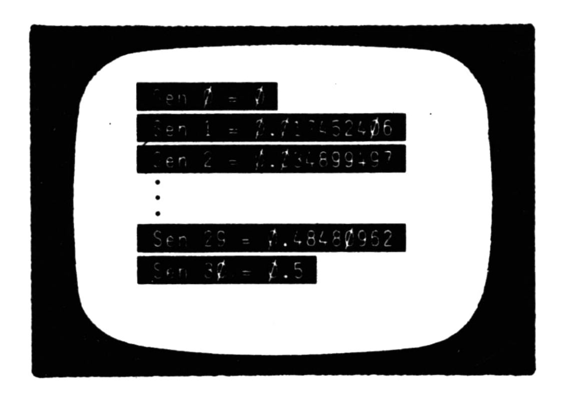
Para escribir los cuadrados de los primeros números impares podemos ejecutar el programa:

```
1Ø  ┌FOR X = 1 TO 19 STEP 2
2Ø  │PRINT "EL CUADRARO DE" ; X ; " ES " ; X↑2
3Ø  └NEXT X
4Ø   END
```

No es necesario que el incremento sea un número entero. Si por ejemplo quisiéramos obtener unas tablas de valores de la función f(x) = sen x desde 0º hasta 30º aumentando cada vez un grado, y teniendo presente que el ordenador trabaja en radianes, pasaremos los grados a radianes y dicha tabla nos la proporciona el siguiente programa:

```
1Ø   REM TABLA DE LA FUNCION SENO DESDE Ø A 3Ø
2Ø  ┌FOR X = Ø TO π/6 STEP π/18Ø
3Ø  │PRINT "Sen " ; X *18Ø/π ; " = " ; SIN(X)
4Ø  └NEXT X
5Ø   END
```

Con la ejecución del mismo aparece en pantalla:



3. Observaciones sobre la sentencia FOR-NEXT

Cuando en un programa aparece FOR X = N TO M se pueden presentar los siguientes casos:

a) Si N = M entonces el bucle se realiza una sola vez.

Ejemplo

```
10  ┌FOR X = 5 TO 5
20  │PRINT X↑3
30  └NEXT X
40   END
```

Con el que aparecerá en pantalla:

125

b) Si N>M y no hay STEP negativo, el bucle se realiza una sola vez.

Ejemplo

```
10  ┌FOR X = 6 TO 2
20  │PRINT X↑3
30  └NEXT X
40   END
```

Con el que aparecerá en pantalla:

216

Nota

En algún ordenador no aparece ningún resultado.

c) Si N>M pero al ir incrementando N mediante el STEP que tenemos en el programa sobrepasa el valor de M, se realiza el bucle hasta el último valor anterior al M.

<u>Ejemplo</u>

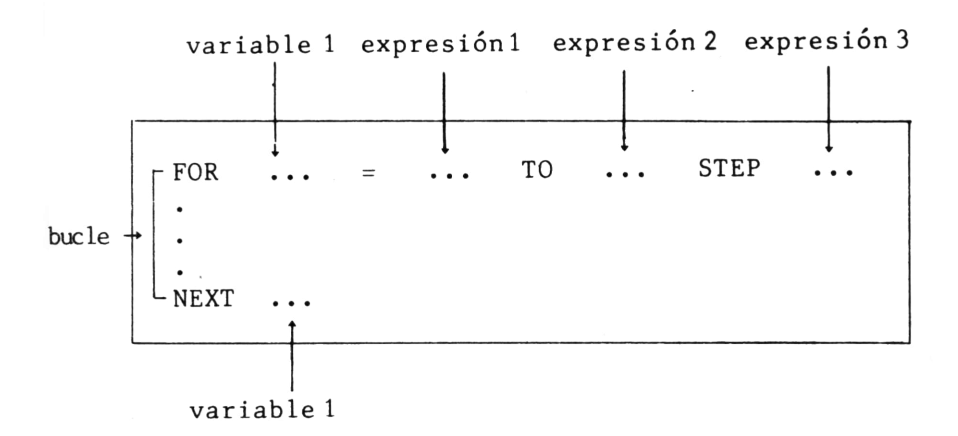
```
10  ┌FOR N = 5 TO 20 STEP 2
20  │PRINT N
30  └NEXT N
40   END
```

En este programa el valor de N va desde el 5 al 19 de 2 en 2 según el STEP que hemos puesto, pero para el siguiente valor, N es igual al 21, es decir, sobrepasa el valor de 20, por lo que finaliza el programa.

Aparece en pantalla:

```
5
7
9
11
13
15
17
19
```

El formato general es:



4. Sentencia PAUSE (pausa)

Con la sentencia PAUSE (n) se detiene la ejecución del programa un tiempo que depende de n.

En algunos ordenadores en vez de PAUSE (n) es WAIT (n).

Se puede también utilizar un bucle para conseguir el mismo fin.

Por ejemplo la línea

```
FOR X = 1 TO N : NEXT N
```

hará una pausa que dependerá del valor de N. En ella hay un bucle en el que le ordenamos que cuente, como cuando de pequeños jugamos al escondite. En otras palabras, le hacemos "perder tiempo".

Puede ser útil una pausa para visualizar algún resultado durante un tiempo determinado y borrarlo después.

EJERCICIOS RESUELTOS

VI.1. *¿Hay algún error en el programa siguiente?*

```
10  ┌ FOR n=1 TO 50
20  │ PRINT n, n+3, n*n
30  │ LET c=5*n
40  └ NEXT c
50    END
```

Solución

Si, puesto que la variable que inicia el bucle es la n mientras que lo cierra la variable c. Daría mensaje de error.

* * *

VI.2. *¿Es correcto el siguiente programa?*

```
10  FOR n=1 TO 10
20  FOR i=1 TO 5
30  NEXT n
40  PRINT n↑i
50  NEXT i
60  END
```

Solución

No pues los bucles están de la siguiente forma:

```
┌
│
│  ┌
└──┼
   │
   └
```

que es incorrecta.

* * *

VI.3. *¿Qué haría la ejecución del siguiente programa?*

```
10  LET S=0
20  INPUT "Escribir un número entero positivo";A
30 ┌FOR n=1 TO A
40 │LET S=S+n↑2
50 └NEXT n
60  PRINT S
70  END
```

Solución

Imprime la suma de todos los cuadrados desde 1^2 hasta A^2.

* * *

VI.4. *Hacer un programa que imprima una tabla de valores de la función*

$$y = \sqrt{e^{\text{sen } x}}$$

desde $x=0$ *hasta* $x=\pi/4$ *aumentando de décima en décima.*

Solución

El programa es:

```
10   REM TABLA DE VALORES
20  ┌FOR X=0 TO π/4 STEP 0.1
30  │PRINT "X= ";X,"Y= ";SQR(EXP(SIN(X)))
40  └NEXT X
50   END
```

* * *

VI.5. *Hacer un programa que imprima una tabla de los números enteros del 1 al 20, sus cuadrados y sus cubos.*

Solución

El programa es:

```
10   PRINT "n";TAB(10);"n↑2";TAB(20);"n↑3"
20   PRINT
30 ┌ FOR n=1 TO 20
40 │ PRINT n; TAB(10);n↑2;TAB(20);n↑3
50 └ NEXT n
60   END
```

* * *

VI.6. *Escribir un programa para calcular n! (usando FOR-NEXT)*

Solución

El programa es:

```
10   REM FACTORIAL DE UN NUMERO
20   INPUT "Número entero positivo";n
30   LET F=1
40 ┌ FOR i=1 TO n
50 │ LET F=F*i
60 └ NEXT i
70   PRINT "El factorial de  ";n;" es  ";F
80   END
```

* * *

VI.7. *Hacer un programa que halle los números de tre cifras* abc *tales que*

$$abc = a^3 + b^3 + c^3$$

Solución

El programa es:

```
10  ┌ FOR a=1 TO 9
20  │  ┌FOR b=∅ TO 9
30  │  │  ┌FOR c=∅ TO 9
40  │  │  │IF 1∅∅*a+1∅*b+c = a*a*a + b*b*b+c*c*c
    │  │  │THEN PRINT  1∅0*a+1∅*b+c
50  │  │  └NEXT c
60  │  └NEXT b
70  └NEXT a
80   END
```

La ejecución del programa da los números:

153, 37∅, 371, 4∅7

* * *

VI.8. *Hacer un programa que halle los números de tres cifras de la forma UNO, DOS tales que verifiquen:*

UNO+UNO = DOS

Solución

El programa es:

```
10  ┌FOR U=1 TO 9
20  │ ┌FOR N=0 TO 9
30  │ │ ┌FOR O=0 TO 9
40  │ │ │ ┌FOR D=1 TO 9
50  │ │ │ │ ┌FOR S=0 TO 9
60  │ │ │ │ │LET X=U*100+N*10+O
70  │ │ │ │ │LET Y=D*100+O*10+S
80  │ │ │ │ │IF 2*X=Y THEN PRINT X,Y
90  │ │ │ │ └NEXT S
100 │ │ │ └NEXT D
110 │ │ └NEXT O
120 │ └NEXT N
130 └NEXT U
140  END
```

La ejecución del programa da los números:

100	200	125	250	150	300
112	224	137	274	162	324
124	248	149	298	174	348
...	...	...	...	...	...

* * *

VI.9. *Escribir un programa que simule el lanzamiento de una moneda y dé como resultado el número de veces que ha salido cara y el de veces que ha salido cruz.*

Solución

El programa es:

```
 10   INPUT "Número de lanzamientos";n
 20   LET ca=0 : LET cr=0
 30  ┌FOR i=1 TO n
 40  │LET x = INT(RND*2)
 50  │IF x=0 THEN LET ca=ca+1 : PRINT "c"
 60  │IF x=1 THEN LET cr=cr+1 : PRINT "+"
 70  └NEXT i
 80   PRINT "Han salido en total:"
 90   PRINT ca;" caras"
100   PRINT cr;" cruces"
120   END
```

EJERCICIOS PROPUESTOS

194. ¿Qué es lo que limita el número de bucles que puede admitir un ordenador en un programa?
a) El número de letras del alfabeto.
b) La capacidad de memoria del ordenador.
c) El que las variables sean numéricas o alfanuméricas.
d) Que el ordenador vaya en pilas o conectado a la red.

195. Si tenemos dos bucles anidados de la forma:

```
┌ FOR n=1 TO 1Ø
│   :
│  ┌ FOR m=2 TO 6
│  │    :
│  └ NEXT m
│   :
└ NEXT n
```

¿cuál de los dos acaba antes?

196. ¿Es posible que en un bucle el valor final de las variables sea menor que el inicial?

*197. ¿Es correcto el programa:

```
1Ø ┌ FOR n=1 TO 1ØØ
2Ø │ LET n=n+3
3Ø │ PRINT n
4Ø └ NEXT n
```

para que n tome los valores desde 1 hasta 1ØØ?

198. Si cambiamos la línea 1Ø por la 2Ø en el programa:

```
1Ø  FOR n=1 TO 1Ø
2Ø  FOR n=1 TO 1Ø STEP 1
```

¿cambiaría el resultado del mismo?

199. ¿Qué imprimirá la ejecución del siguiente programa?

```
 1Ø ┌FOR A=23 TO 5 STEP -2
 2Ø │PRINT "A = ";A+1
 3Ø │IF A=15 GO TO 1ØØ
 4Ø └NEXT A
1ØØ  END
```

* 200. ¿Qué imprimirá la ejecución del siguiente programa?

```
1Ø  INPUT X
2Ø  IF X>6 THEN GO TO 4Ø
3Ø ┌FOR n=1 TO 1Ø
4Ø │PRINT X
5Ø └NEXT n
6Ø  PRINT X+4
7Ø  END
```

201. ¿Qué imprimirá la ejecución del siguiente programa?

```
1Ø ┌FOR n=4 TO 1
2Ø │PRINT n
3Ø └NEXT n
4Ø  END
```

202. ¿Qué hará la ejecución del siguiente programa?

```
1Ø ┌FOR n=5 TO 1Ø STEP 1Ø
2Ø │PRINT n
3Ø └NEXT n
4Ø  END
```

203. En el supuesto de que el ordenador tenga 4 zonas de impresión en pantalla ¿qué imprimirá el siguiente programa?

```
10 ┌FOR n=1 TO 20
20 │PRINT, N,
30 │  ┌FOR I=2 TO 3
40 │  │PRINT N ↑(1/I),
50 │  └NEXT I
60 └NEXT N
70  END
```

*204. ¿Qué ocurrirá al ejecutar el siguiente programa?

```
10 ┌FOR n=1 TO 20
20 │LET n=n-1
30 │PRINT n
40 └NEXT n
50  END
```

205. ¿Hay algún error en el siguiente programa?

```
10 ┌FOR X=5 TO -3 STEP -1
20 │LET Y=X↑2
30 │PRINT Y
40 └NEXT X
50  END
```

206. ¿Hay algún error en el siguiente programa?

```
10  FOR X=1 TO 10
20  PRINT X
30  FOR Y=1 TO 5
40  PRINT X*Y
50  NEXT X
60  NEXT Y
70  END
```

207. ¿Hay algún error en el siguiente programa?

```
10  LET h=1.4
20 ┌FOR n=1 TO 10 STEP h
30 │PRINT n*h
40 └NEXT h
50  END
```

208. ¿Hay algún error en el siguiente programa si se pretende escribir del 1 al 20?

```
10 ┌FOR X=1 TO 20
20 │PRINT X
30 │GO TO 10
40 └NEXT X
```

¿Qué escribirá?

209. ¿Hay algún error en el siguiente programa?

```
10  FOR X=1 TO 10
20  FOR Y=3 TO 8
30  LET Z=X+Y
40  NEXT X
50  PRINT Z
60  NEXT Y
70  END
```

210. ¿Hay alguna manera de salir del bucle siguiente

```
10 ┌FOR n=1 TO 200
20 │PRINT "Seguro"
30 └NEXT n
```

añadiéndole alguna instrucción?

* 211. Escribir un programa equivalente al dado utilizando la sentencia FOR-NEXT.

```
10  LET X=3
20  PRINT X↑2+1
30  LET X=X+1
40  IF X<=20 THEN GO TO 20
50  PRINT "FIN"
60  END
```

212. Hacer un programa cuya ejecución escriba 50 veces:

QUIEN SIEMBRA VIENTOS RECOGE TEMPESTADES,

utilizando la sentencia FOR-NEXT.

213. Hacer un programa cuya ejecución imprima todos los cubos y las raíces cúbicas de 1 a 100 utilizando la sentencia FOR-NEXT.

214. Hacer un programa cuya ejecución imprima una tabla de valores de una función entre 0 y n (n entero) de décima en décima.

215. Escribir un programa cuya ejecución calcule la suma de los 50 primeros números impares utilizando la sentencia FOR-NEXT.

216. Escribir un programa cuya ejecución calcule:

$$S = 1 + \frac{1}{2} + \frac{1}{2^2} + \dots + \frac{1}{2^n}$$

entrando n mediante la sentencia INPUT.

217. Escribir un programa cuya ejecución pida las notas de matemáticas de 30 estudiantes y dé como resultado la media aritmética.

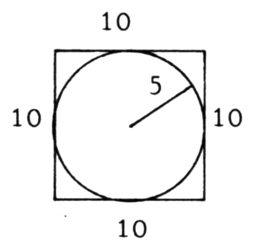
*218. Dada la progresión geométrica:

$$\begin{cases} a_1,\ a_1 r,\ a_1 r^2,\ a_1 r^3,\ \ldots \\ |r| < 1 \end{cases}$$

hacer un programa cuya ejecución halle la suma de los N primeros términos y compare dicha suma con el límite que es

$$a_1/(1-r)$$

*219. Dada la figura adjunta



hacer un programa que dé N puntos al azar dentro del cuadrado, que calcule la frecuencia relativa de los puntos que están dentro de la circunferencia y que imprima esta frecuencia relativa multiplicada por 4.

Observar que el resultado así obtenido es una aproximación de π.

(Los puntos (x,y) interiores a la circunferencia verifican:

$$\sqrt{(x-5)^2+(y-5)^2} < 5 \quad).$$

220. Escribir un programa para comprobar si existe algún número de tres cifras de la forma abc tal que sea igual a la suma de los cuadrados de sus cifras.

221. La sucesión de Fibonacci tiene de término general

$$a_n = a_{n-1} + a_{n-2} ,$$

siendo $a_1=0$ y $a_2=1$. Escribir un programa que nos dé cualquier término de esta sucesión a partir del 3º.

222. Escribir un programa que nos dé todos los números de 4 cifras en base 4 y su equivalente en el sistema decimal.

223. Escribir un programa que simule el lanzamiento de 100 veces un dado y que escriba el número de veces que ha salido cada cara.

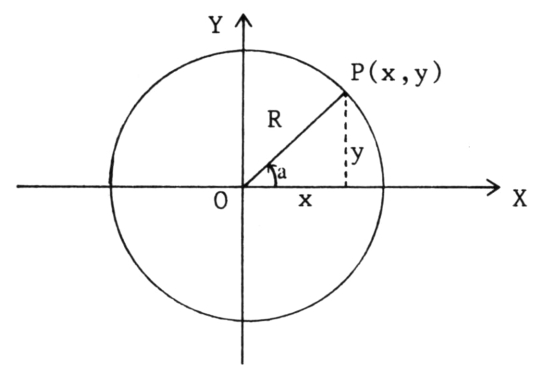
*224. Esecribir un programa para hallar los números enteros A,B,C tales que $C^2=A^2+B^2$ (números Pitagóricos) del 1 al 40.

225. Hacer un programa que escriba todos los números de dos cifras en base 5 y sus equivalentes en base 10.

226. Hacer un programa para calcular $V_{m,n}$ siendo

$$V_{m,n} = m(m-1) \ldots (m-n+1)$$

*227. Hacer un programa para hallar las coordenadas de n puntos de la circunferencia de ecuación $x^2+y^2=R^2$ que sean vértices de un polígono regular.



Las coordenadas (x,y) de P vienen dadas por

$$\begin{cases} x = R \cos a \\ y = R \operatorname{sen} a \end{cases}$$

228. Hacer un programa que imprima todos los resultados posibles de sumar los cuadrados de los números obtenidos al echar 3 dados.

*229. Hacer un programa para averiguar si hay algún capicúa de tres cifras que sea igual a la suma de los cuadrados de las mismas.

230. Hacer un programa para averiguar si hay números enteros entre 1 y 100 que verifiquen la igualdad

$$a^3 = b^3 + c^3$$

(caso particular del teorema de Fermat).

231. Hacer un programa que halle los números de tres cifras de la forma RIO, MAR tales que verifiquen:

RIO + RIO = MAR

232. Hacer un programa que halle los números de 4 cifras de la forma MIAU, GATO tales que verifiquen:

MIAU + MIAU = GATO

233. Hacer un programa que halle los números de la forma HIP, HURRA tales que verifiquen:

HIP * HIP = HURRA

234. Hacer un programa que averigue si hay números de la forma PIO, POLLO tales que verifiquen:

PIO * PIO = POLLO

*235. Escribir un programa para hallar todos los divisores (primos o no) de un número dado.

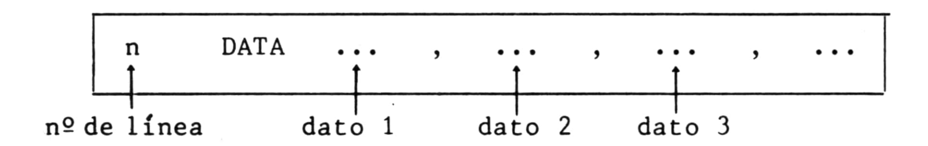
7
ALMACENAMIENTO DE DATOS

Con la sentencia INPUT entramos datos desde fuera del programa. Si estos datos son numerosos y conocidos de antemano es más cómodo tenerlos almacenados en una línea de programa.

Esto se consigue con la sentencia DATA.

1. Sentencias DATA (datos) y READ (leer)

Con la sentencia DATA podemos almacenar distintos datos en un programa, ya sean numéricos o alfanuméricos. Para ello bastará poner: DATA seguido de los datos, separados por comas, con el siguiente formato:



Ejemplo DATA 5,7,"JUANA",18,85

Para utilizar los distintos datos almacenados con la sentencia DATA se utiliza la sentencia READ.



Ejemplo

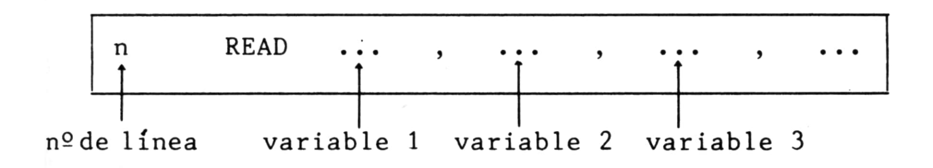
```
1Ø  DATA 5,7,"Juana",18,85
2Ø  READ X,Y
3Ø  READ A$,Z,T
4Ø  PRINT Y,X
5Ø  PRINT T,A$,Z
6Ø  END
```

La ejecución del programa da como resultado:

```
7           5
85          JUANA       18
```

Vemos pues que la misión del READ es leer los datos almacenados y asignarlos a unas variables.

El formato es:



Cuando en el programa aparece por primera vez

una sentencia READ busca el primer dato almacenado en la primera sentencia DATA y lo asigna a la primera variable de READ y así sucesivamente ..., o sea:

variable 1 = dato 1
variable 2 = dato 2
variable 3 = dato 3
....

Cuando aparece una nueva sentencia READ asigna a sus variables los datos a partir del último leído anteriormente (Es como si los datos se fueran gastando).

Siguiendo el orden de lectura las variables numéricas deben corresponder a datos numéricos y las alfanuméricas a cadenas de caracteres de lo contrario habrá mensaje de error.

Al usar pues las sentencias DATA y READ tendremos presente que se leerán los datos numéricos o alfanuméricos en riguroso orden, esto es así aún empleando el mismo nombre de la variable, como veremos en el siguiente ejemplo:

Ejemplo

```
 1Ø  DATA 1,2,3,4,5,6
 2Ø  READ X,Y,Z
 3Ø  PRINT X,Y,Z
 4Ø  READ X,Y,Z
 5Ø  PRINT X,Y,Z
1ØØ  END
```

La ejecución de este programa da:

```
1     2     3
4     5     6
```

Puede ocurrir que se acabe el programa sin haberse leído todos los datos de la sentencia o sentencias DATA, con lo cual no pasa nada, solamente que había una superabundancia de datos almacenados.

Si se acaban los datos almacenados y quedan variables en READ aparece mensaje de error.

Si en el programa anterior añadimos las líneas:

```
60  READ X
70  PRINT X
```

en la ejecución del programa aparece entonces

```
1     2     3
4     5     6
Mensaje de error.
```

Veamos otro caso en el que aparece mensaje de error por falta de datos en la sentencia DATA. Puede ocurrir fácilmente cuando aparecen las sentencias DATA y READ en un bucle como en el ejemplo:

```
10   DATA 5,7,12,15,20
20   READ X
30   PRINT X,2*X
40   GO TO 20
50   END
```

La ejecución de este programa da:

```
 5          10
 7          14
12          24
15          30
20          40
Mensaje de error
```

Este error es debido a que se ha intentado leer un dato que no existe.

Podemos salirnos de este bucle sin necesidad de que nos señale un error, cosa que por otra parte nos detendría el programa (y estas líneas pueden ser solamente parte de un programa más extenso). Para que no aparezca este error en la pantalla, se suele

dar en los datos de DATA un valor "ficticio", que al tomarlo la variable correspondiente se efectúa un salto en el programa. En el ejemplo anterior podemos poner un cero en el último valor de DATA e intercalar un condicional de la siguiente manera:

```
10  DATA 5,7,12,15,20,0  ← valor ficticio
20  READ X
25  IF X=0 THEN GO TO 50
30  PRINT X,2*X
40  GO TO 20
50  PRINT "MUY BIEN"
:
```

Aparece en pantalla:

```
 5      10
 7      14
12      24
15      30
20      40
MUY BIEN
..........
```

Si tenemos que realizar un programa para calcular la media aritmética de una serie de números, todos ellos positivos, es cómodo usar como valor "ficticio" un número negativo, por ejemplo -1.

Ejemplo

```
 5  REM NOTA MEDIA
10  LET M = 0
20  DATA 7 , 4.2 , 2.8 , 6 , 5 , 1 , 9 , 8.5 ,
    2.6 , 5 , 4 , 2 , 6 , 7 , 6.5 , 4.5 , 3 ,
    7 , 5 , 5.8 , 10 , 4 , 2 , 7 , 6 , -1
30  READ X                                 ↑
40  IF X = -1 THEN GO TO 70          valor ficticio
50  LET M = M + X
60  GO TO 30
70  PRINT "LA NOTA MEDIA ES M = " ; M/25
80  END
```

Este programa nos da en pantalla:

LA NOTA MEDIA ES M = 5.236

Nota 1:

Puede haber más de una sentencia DATA en un mismo programa, por ejemplo:

```
10  DATA 7 , 9 , "Amarillo" , "Violeta"
20  DATA "Negro" , 16 , 50
```

Su acción es la misma que la única sentencia:

```
15  DATA 7 , 9 , "Amarillo" , "Violeta" , "Ne-
    gro" , 16 , 50
```

Nota 2:

No es necesario que las sentencias DATA y READ vayan seguidas como han aparecido en los ejemplos. Es cómodo colocar las sentencias DATA todas al principio del programa o bien al final del mismo.

2. Sentencia RESTORE (restaurar)

Al usar la sentencia READ "gastábamos los datos almacenados en DATA. Si nos interesa emplearlos otra vez en el programa, nos basta escribir en el mismo la sentencia RESTORE, cuya misión es realmacenar de nuevo todos los datos de todas las sentencias DATA del programa, de manera que el siguiente READ volverá a leer desde el principio.

Veámoslo en el siguiente ejemplo:

```
 5  REM USO DE RESTORE
10  DATA 3 , 5 , 7 , -1  <-- valor ficticio
20  READ X
30  IF X = -1 THEN GO TO 60
```

```
 40  PRINT X , X↑0.5
 50  GO TO 20
 60  RESTORE           (vuelven a estar almacenados
 70  READ X             los números de la línea 10)
 80  IF X = -1 THEN GO TO 110
 90  PRINT X , X↑2
100  GO TO 70
110  PRINT "FIN DEL EJEMPLO"
```

Con este programa aparece en la pantalla:

```
3       1.73205
5       2.23607
7       2.64575
3             9
5            25
7            49
FIN DEL EJEMPLO
```

EJERCICIOS RESUELTOS

VII.1. *¿Saldrá mensaje de error al ejecutar el siguiente programa?*

```
10  DATA 1 , 3 , 4 , 7 , 8 , 10 , 11 , 7
20 ┌FOR n = 1 TO 6
30 │READ a : PRINT a
40 └NEXT n
50  END
```

Solución:

No, pues el hecho de que no se utilicen todos los números almacenados en un DATA no produce mensaje de error.

* * *

VII.2. *¿Qué imprimirá la ejecución del siguiente programa?*

```
10 ┌FOR i = 1 TO 4
20 │READ X
30 │PRINT X
40 │RESTORE
50 └NEXT i
60  DATA 5,7,15,21
70  END
```

Solución

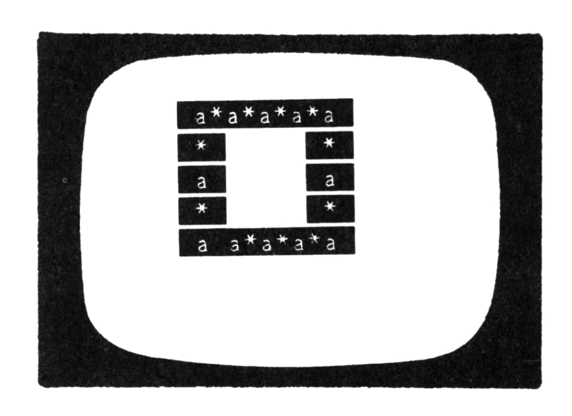
Debido a la línea 40 el READ sólo leerá el primer dato 5. Imprimirá 4 veces 5.

* * *

VIII.3. *¿Qué imprimirá la ejecución del siguiente progra ma?*

```
 10  READ X
 20  IF  X=1  THEN GO TO 80
 30  IF  X=2  THEN GO TO 100
 40  IF  X=3  THEN GO TO 120
 50  IF  X=4  THEN GO TO 100
 60  IF  X=5  THEN GO TO 80
 70  IF  X=0  THEN GO TO 150
 80  PRINT TAB(11); "a*a*a*a*a"
 90  GO TO 10
100  PRINT TAB(11); "*";TAB(19);"*"
110  GO TO 10
120  PRINT TAB(11);"a";TAB(19);"a"
130  GO TO 10
140  DATA 1,2,3,4,5,0
150  END
```

Solución



* * *

VIII.4. *Hacer un programa que introduciendo los siguiente números: 2,14,23,54,60,37,125,8,246,29*

en una sentencia DATA imprima sólo los pares.

Solución

El programa es:

```
10 ┌FOR i=1 TO 10
20 │READ A
30 │IF A/2 = INT(A/2) THEN PRINT A,
40 └NEXT i
50  DATA 2,14,23,54,60,37,125,8,246,29
60  END
```

* * *

VII.5. *En la sentencia DATA de un programa se van anotando los datos 1,2,3 según las respuestas de una cierta encuesta. Si hay 67 datos ¿cómo debe ser dicho programa para que al final imprima los tantos por cientos de cada resultado?.*

Solución

El programa es:

```
 10  LET A1=0 : LET A2=0 : LET A3=0
 20 ┌FOR i=1 TO 67
 30 │READ X
 40 │IF X=1 THEN LET A1=A1+1
 50 │IF X=2 THEN LET A2=A2+1
 60 │IF X=3 THEN LET A3=A3+1
 70 └NEXT i
 80  PRINT "Del 1 hay  ";A1*100/67;" %"
 90  PRINT "Del 2 hay  ";A2*100/67;" %"
100  PRINT "Del 3 hay  ";A3*100/67;" %"
110  DATA 1,3,2,3,1,1,2,2,3,1,...(hasta 67 datos)
120  END
```

* * *

VII.6. *Se introduce en un programa las edades de un conjunto de N personas mediante la sentencia DATA. Completar el programa para que dé:*

a) El número de personas cuya edad esté comprendida entre 18 y 28 años (ambos incluídos).
b) El número de personas de edad superior a 50 años.
c) La edad media.

Solución

El programa es:

```
 10  LET A=0
 20  LET B=0
 30  LET T=0
 40 ┌FOR i=1 TO N
 50 │READ X
 60 │LET T=T+X
 70 │IF X>=18 AND X<=28 THEN LET A=A+1
 80 │IF X > 50 THEN LET B=B+1
 90 └NEXT i
100  PRINT "Número de personas de edad
     entre 18 y 28 = " ; A
110  PRINT "Número de personas de edad
     mayor que 50 = " ; B
120  PRINT "Edad media = "; T/N
130  DATA ... (Edades de las N personas)
140  END
```

* * *

VII.7. *Hacer un programa que tenga almacenados con DATA los nombres de las estaciones de cada línea de metro de Barcelona y que introduciendo el número de la línea dé como resultado todas las estaciones de la misma en orden, con los enlaces respectivos.*

Solución

El programa es:

```
10  INPUT "Qué línea desea consultar";A
20  PRINT "Las estaciones de la línea   ";A;"son:".
    PRINT
30 ┌FOR i=1 TO 82
40 │READ N, E$
50 │IF N=A THEN PRINT E$
60 └NEXT i
70  DATA 1, "Torrasa",1,"Santa Eulalia",1,"Mercat Nou",
    1,"P. de Sants (enlace con línea 5)",1,"Hostafrancs'
    1,"Espanya (enlace con línea 3)",1,"Rocafort",1,
    "Urgell",1,"Universitat",1,"Catalunya (enlace cor
    línea 3)",1,"Urquinaona(enlace con línea 4)"
80  DATA 1,"Arc de Triomf",1,"Marina",1,"Glorias",1,
    "Clot",1,"Navas",1,"Sagrera (enlace con línea 5),1,
    "Fabra y Puig",1,"Sant Andreu",1,"Torras y Bages",1,
    "Trinitat Vella",1, "Baró de Viver",1,"Santa Coloma"
90  DATA 3,"Zona Universitària",3,"Palau Reial",3,"Maria
    Cristina",3,"Les Corts",3,"P. del Centre",3,"Sants-
    estació (enlace con lína 5)",3,"Espanya (enlace
    con lína 1)",3,"Poble Sec",3,"Paral.lel",3,"Drassa-
    nes",3,"Liceu",3,"Catalunya (enlace con línea 1)"
100 DATA 3,"Passeig de Gràcia (enlace con línea 4)",3,
    "Diagonal (enlace con línea 5)",3,"Fontana",3,"Les-
    seps",4,"Roquetes",4,"Llucmajor",4,"Maragall (enlace
    con línea 5)",4,"Guinardó",4,"Alfons X",4,"Joanich"
110 DATA 4,"Verdaguer (enlace con línea 5)",4,"Girona",
    4,"Passeig de Gràcia (enlace con línea 3)",4,"Urqui-
    naona (enlace con línea 1)",4,"Jaume I",4,"Barcelo-
    neta",4,"Ciutadella",4,"Bogatell",4,"Llacuna",4,"Po-
    blenou"
120 DATA 4,"Selva de Mar",4,"Mina",4,"Bessós",4,"La
    Pau",5,"Cornellà",5,"Gavarra",5,"Sant Ildefons",5,
    "Can Boixeres",5,"Can Vidalet",5,"Pubilla Cases",5,
    "Collblanch",5,"Badal",5,"P.de Sants (enlace con lí-
    nea 1)"
130 DATA 5,"Sants-estació (enlace con línea 3)",5,"Enten
    ça",5,"Hospital Clinic",5,"Diagonal (enlace con
    línea 3)",5,"Verdaguer (enlace con línea 4)",5,"Sa-
    grada Familia"
140 DATA 5,"Hospital de Sant Pau",5,"Camp del arpa",5,
    "Sagrera (enlace con línea 1)",5,"Congrès",5,"Mara-
```

```
    gall (enlace con línea 1)",5,"Virrei Amat",5,"Vila-
    piscina",5,"Horta"
150 END
```

* * *

VII.8. *Escribir un programa que teniendo almacenados con DATA el Nombre, Edad, Estado civil, número de hijos y sueldo de los empleados de una empresa imprima el nombre de aquellos que su edad esté comprendida entre 22 y 35 años (ambos inclusive) y son solteros.*

Solución

Siendo N el número de empleados, el programa es:

```
10 ┌FOR I=1 TO N
20 │READ N$,E,C$,F,S
30 │IF(E>=22 AND E<=35) AND C$="S" THEN
   │PRINT N$
40 └NEXT I
50  DATA "José García Pérez", 23,"S",Ø,
    8ØØØØ,... (hasta N empleados)
```

Nota S significa: soltero
C significa: casado

* * *

VII.9. *Con los mismos datos del problema anterior modificar el programa de manera que imprima la lista de los nombres de los casados con más de 2 hijos y con sueldo inferior a 80000 ptas.*

Solución

En el programa anterior basta cambiar la línea 3Ø por:

```
30  IF C$="C" AND F>2 AND S<8ØØØØ THEN PRINT N$
```

* * *

EJERCICIOS PROPUESTOS

236. Si en un programa hay n líneas con la sentencia READ ¿deben haber forzosamente n líneas con la sentencia DATA?

237. ¿Pueden mezclarse en una línea DATA cantidades numéricas y cadenas de caracteres?

238. La entrada de datos en un programa podemos efectuarla mediante las sentencias INPUT o DATA. ¿Cuándo emplearemos una u otra?

239. ¿En qué orden hay que escribir las sentencias READ y DATA?

a) Necesariamente READ antes que DATA.
b) Necesariamente DATA antes que READ.
c) Es indistinto el orden para la ejecución del programa.
d) Necesariamente la DATA al final y READ al principio del programa.

Razonar la respuesta correcta.

240. ¿Es cierto que una vez utilizado un número almacenado en un DATA, nunca puede volver a utilizarse?

*241. ¿Hay algún error en el siguiente programa?

```
10  READ B,C
20  PRINT B,C
30  READ A$ , B$
40  PRINT A$ , B$
50  DATA 1,2,3,"libro" , "estudio" , 7
60  END
```

242. ¿Hay algún error en el siguiente programa?

```
10  READ A,B,C
20  PRINT "La media aritmética es"
30  PRINT (A+B+C) / 3
40  END
```

*243. En el siguiente programa señalar los errores y corregirlos.

```
10  READ A ; B
20  LET A + B = X
30  PRINT X
40  DATA 23 ; 10
50  END
```

244. ¿Hay algún error en el siguiente programa?

```
10  READ X,Y,Z
20  LET A = X - 2 * Y + 4 * Z
30  DATA 37 , 12 , 24.5 , 148
```

245. Hallar los errores que hay en los siguientes programas:

a)
```
10  READ X ; Y , Z
20  PRINT X , Y , Y+Z
30  DATA 1 , 3.5 , 72
40  END
```

b)
```
10  READ A$ , A , B
20  PRINT A$
30  PRINT A - B
40  DATA "JUAN" , "JORGE" , 7
50  END
```

246. ¿Hay algún error en el siguiente programa?

```
10  READ A,B,C,D,E
20  LET S = A+B+C+D+E
30  PRINT S
40  DATA 24,12,37,5.83
50  END
```

247. ¿Qué valores se asignan a las variables A,B,C,D,E,F al ejecutar el siguiente programa?

```
10  DATA 3,-1,5,7E2
20  READ A,C,E
30  READ F
40  RESTORE
50  READ B,D
60  END
```

*248. ¿Qué imprimirá la ejecución del siguiente programa?

```
10  READ A,B,C
20  LET X = B-C
30  RESTORE
40  READ X,Y,Z,T
50  LET Y = A + 2 * Z
60  PRINT A,B,C,X,Y,Z,T
70  DATA 2,4,-3,7.2
80  END
```

249. ¿Qué imprimirá la ejecución del siguiente programa?

```
10  LET n = 1
20  READ a
30  PRINT a*a,
40  LET n = n + 1
50  IF n< 4 THEN GO TO 20
60  PRINT "se acabó"
70  DATA 1,4,6,10,20,25,32,40
80  END
```

250. ¿Qué imprimirá la ejecución del siguiente programa?

```
10  ┌ FOR n = 1 TO 5
20  │ READ A , B$
30  │ PRINT A , B$
40  │ RESTORE
50  └ NEXT n
60    DATA 5,"Antonio",4,"Luisa",8,"Ana",6,
      "David",5,"Anacleto"
70    END
```

251. ¿Qué imprimirá la ejecución del siguiente programa?

```
10  READ X,Y
20  PRINT X,Y
30  DATA 2.5,3.4
40  READ Z
50  PRINT Z ↑ 2 + X ↑ 2 + Y ↑ 2
60  END
```

*252. Explicar la finalidad del siguiente programa.

```
10    LET S = 0
20  ┌ FOR n = 1 TO 7
30  │ READ A$ , B
40  │ LET S = S + B
50  │ PRINT "El  " ; A$ ; "  se registró  " ;
    │ B ; "  grados"
55  │ PRINT
60  └ NEXT n
```

```
65  PRINT
7Ø  DATA "Lunes",2Ø,"Martes",18,"Miércoles",19,
    "Jueves",22,"Viernes",19,"Sábado",21,"Domin
    go",2Ø
8Ø  PRINT "la temperatura media de la semana ha
    sido:  ";S/7
9Ø  END
```

253. Escribir un programa equivalente al

```
1Ø  LET A=5 : PRINT A,
2Ø  LET B=15: PRINT B,
3Ø  LET C=8 : PRINT C,
4Ø  LET D=25: PRINT D,
5Ø  LET E=3Ø: PRINT E,
6Ø  PRINT "La media geométrica es:  ";(A*B*C*D*E)
    ↑(1/5)
7Ø  END
```

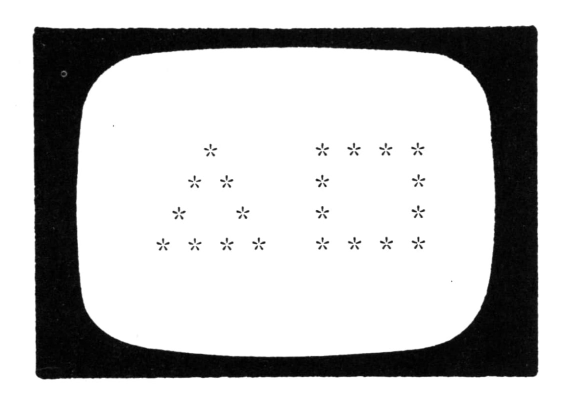
utilizando las sentencias READ y DATA.

*254. Escribir un programa cuya ejecución imprima 20 veces los cuatro primeros números almacenados en una línea DATA con 7 números.

255. Escribir un programa empleando la sentencia DATA para asignar los valores 3.10^{-8}, sen 34º, arc sen 0,34 a las variables A1, A2 y A3.

256. Escribir un programa empleando una sentencia READ cuya ejecución imprima los logaritmos neperianos de los números 23, 135, 45.7, 28, 147, 1412, 1936.

257. Escribir un programa cuya ejecución imprima el gráfico adjunto utilizando las sentencias READ y DATA.



*258. Escribir un programa en el que mediante la sentencia DATA se introduzcan los gastos de cada día de la semana y que calcule el gasto total y el gasto medio.

259. En diversas observaciones de un fenómeno se han conseguido las siguientes cantidades:

$x_1=2,33$ $\quad x_4=5,01$
$x_2=5,12$ $\quad x_5=4,91$
$x_3=4,11$ $\quad x_6=5,25$

Escribir un programa en el que se calcule:

a)
$$m = \frac{\sum_{i=1}^{6} x_i}{6} \quad \text{(media)} \quad \left(\sum_{i=1}^{6} x_i = x_1+x_2+\ldots+x_6\right)$$

b)
$$\frac{\sum_{i=1}^{6} x_i^2}{6}$$

utilizando las sentencias READ y DATA.

*260. Se introducen en un programa mediante la sentencia DATA las alturas de un conjunto de N personas. Completar el programa para que dé:
a) El número de personas cuya altura sea mayor que 1,70 m.
b) El número de personas cuya altura sea menor que 1,60 m.

261. Se introducen en un programa mediante la sentencia DATA las temperaturas de cada día del mes de diciembre. Completar el programa para que dé:
a) El número de días en que la temperatura es menor que 5º
b) El número de días en que la temperatura es mayor que 10º.
c) La temperatura media.

262. Escribir un programa que almacene las edades de un conjunto de 20 personas y luego dé como resultado el número de personas con edad menor o igual que 15, el número que hay entre 15 y 30, entre 30 y 45 y mayores de 45 años. Aplicarlo al caso en que las edades sean: 10,20,17,24,35,32,54,43,8,27,38,19,35,7,56,61,15,42,48 y 14.

263. Completar el programa del ejercicio anterior para que dé además la edad media y la desviación típica.

$$\left(\text{media} = m = \frac{\sum_{i=1}^{n} x_i}{n},\quad \text{desviación típica} = \sigma = \sqrt{\frac{\sum_{i=1}^{n} (x_i - m)^2}{n}}\right)$$

*264. Escribir un programa que almacene 10 nombres en una línea DATA y cuya ejecución imprima el primero

de ellos en orden alfabético. Aplicarlo al caso de que los nombres sean: PEDRO, JUAN, JORDI, XAVIER, DAMIAN, DAVID, PEPE, LUIS, ORIOL y MANOLO.

*265. Escribir un programa que imprima el siguiente laberinto empleando las sentencias READ y DATA.

```
*  ********
*  ****   *
*     * ***
**  * * ***
**  * * ***
**  *     *
**  *** * *
**  *** * *
*   **  * *
********* *
```

266. Hacer un programa para que imprima el número de aprobados y suspensos de una asignatura de un curso de 40 alumnos almacenando las notas con DATA (Es aprobado con nota mayor o igual que 5).

267. Hacer un programa que tenga almacenados con DATA los nombres de los elementos químicos con sus símbolos y que introduciendo el nombre de un elemento dé como respuesta el símbolo del mismo.

268. Hacer un programa que tenga almacenados con DATA las naciones de Europa y sus capitales, y que introduciendo el nombre de una nación dé como respuesta la capital de la misma.

*269. Hacer un programa que tenga almacenados los nombres de naciones de América del Sur con sus capitales y que el ordenador pregunte al usuario la capital de una cierta nación, en caso de no acertar que pruebe

de nuevo hasta 3 intentos y si acierta que el ordenador pregunte de nuevo la capital de otra nación hasta el final de la lista.

270. Hacer un programa que tenga almacenadas las fórmulas de las áreas de las figuras geométricas elementales y que introduciendo el nombre de una figura dé como respuesta el área respectiva.

*271. Hacer un programa que tenga almacenados los premios Nobel de 1970 a 1980 (ambos incluidos) y que introduciendo el nombre de la materia dé como resultado los nombres de todos los premiados en esta década.

272. Hacer un programa que tenga almacenados con DATAS los nombres de entremeses, pescados, carnes y postres con su precio, de un restaurante y que introduciendo el número 1 nos dé los distintos entremeses, el 2 los distintos pescados, el 3 las distintas carnes y el 4 los distintos postres.

273. Hacer un programa que tenga almacenados los nombres de un conjunto de escritores y algunas de sus obras más representativas y que introduciendo el nombre de un escritor dé como respuesta estas obras.

274. Hacer un programa que tenga almacenados los nombres de cierto número de personas con sus respectivos teléfonos y que introduciendo el nombre de una persona dé como respuesta su número de teléfono.

8
SUBRUTINAS

Si en un programa hay una parte del mismo que se repite en varios lugares, es cómodo considerarlo como un subprograma a fin de que sólo se escriba una vez. Estos subprogramas se llaman <u>subrutinas</u>. Esto se consigue utilizando las sentencias

GOSUB y RETURN

1. Sentencias GOSUB(ir a la subrutina) y RETURN(volver)

La sentencia GOSUB se escribe de la forma siguiente:

m GOSUB k

donde m es el número de la línea de la sentencia GOSUB y k es el número de la 1ª línea de la subrutina.

GOSUB k actúa en principio como GO TO k, con la diferencia esencial de que el ordenador recuerda el número de línea m, de tal forma que una vez ejecutada la subrutina, esto es, cuando en la subrutina encuentra RETURN, vuelve a la línea siguiente de m.



La sentencia RETURN se escribe de la forma siguiente:

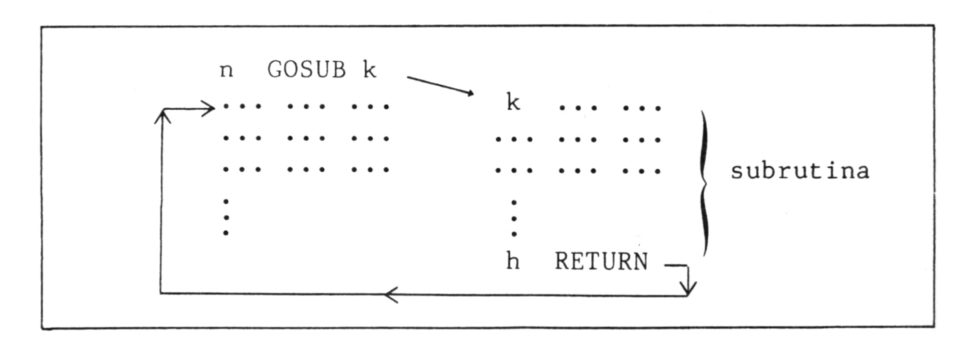
h RETURN

Esta sentencia sirve para señalar el final de una subrutina.

Cuando debido a un GOSUB se está ejecutando una subrutina, RETURN tiene la misión de hacer volver la ejecución del programa a la línea siguiente de este GOSUB que lo ha mandado a la subrutina.

Toda subrutina debe acabar pues con un RETURN y es útil empezarla con una sentencia REM que indique la función de la subrutina.

El formato es:



Ejemplos

a) Supongamos que en varios lugares de un programa interesa calcular valores de la función:

$$y = x^2 + \text{sen } x - 1$$

El cálculo de estos valores lo escribiremos en forma de la subrutina siguiente:

```
5ØØ  REM CALCULO DE VALORES DE LA FUNCION
     y = x↑2 + sen x - 1
51Ø  LET Y=x*x+SIN(x)-1
52Ø  RETURN
```

Cada vez que en el programa principal se quiera calcular $y = x^2 + \text{sen } x - 1$ bastará escribir GOSUB 5ØØ con lo cual y será el valor de la función.

b) Supongamos que en varios lugares de un programa interesa que de tres variables a,b,c se elija la menor asignando su valor a la variable m. Para ello utilizaremos la siguiente subrutina:

```
5ØØ  REM SUBRUTINA QUE ASIGNA A m EL VALOR MENOR
     DE a,b,c
51Ø  IF  b< a  THEN LET a=b
52Ø  IF  c < a  THEN LET a=c
53Ø  LET m=a
54Ø  RETURN
```

Cada vez que en el programa principal aparezca GOSUB 5ØØ a partir de la línea siguiente del GOSUB m será el menor de los números a,b,c

c) El siguiente programa:

```
1Ø  ┌ FOR I=1 TO 6
2Ø  │ READ x,z
3Ø  │ GOSUB  1ØØ
4Ø  └ NEXT I
```

```
100  REM AVERIGUA SI EL PUNTO ES DE LA CURVA
110  LET y=x*x+2*x+4
120  IF y<> z  THEN RETURN
130  PRINT "(";x;",";z;")";"  es un punto de la
     curva"
140  RETURN
150  DATA 1,7,2,3,0,4,4,7,8,84,14,17
160  END
```

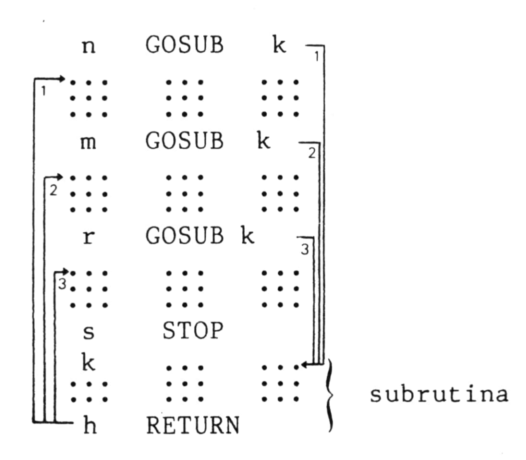
averigua si seis puntos dados pertenecen a la curva de ecuación:

$$y = x^2 + 2x + 4$$

Notas

a) Las subrutinas suelen colocarse al final del programa (aunque esto no es necesario). Es útil preceder las subrutinas con una sentencia STOP o bien GO TO hacia fuera de la subrutina para que sólo se ejecute a continuación de un GOSUB.

Un programa con varias llamadas a una subrutina lo podemos representar esquemáticamente:



b) Dentro de una subrutina puede haber varias sentencias RETURN (formando parte de sentencias IF...THEN...).

El primer RETURN que encuentra devuelve la ejecución del programa a la línea siguiente del GOSUB.

Ejemplo

La subrutina:

```
300  IF a/2=INT(a/2) THEN RETURN
310  LET a=2*a
320  RETURN
```

multiplicará los números impares por 2 y no alterará los números pares.

2. Funciones definidas por el usuario

En el TEMA II vimos que podíamos usar funciones matemáticas que el ordenador tiene ya incorporadas en la memoria ROM, pero si necesitamos alguna función que no esté definida allí lo podemos conseguir con la sentencia DEF FN cuya misión es precisamente definir funciones que necesita el usuario.

La sentencia DEF FN va seguida de una letra que es el nombre de la función y otra u otras entre paréntesis para las variables.

Por ejemplo la función $y(x) = x^2-5x+4$ la definiríamos en BASIC de la forma:

```
DEF FN y(x) = x*x - 5*x + 4
```

Para obtener valores de una función definida se utiliza la sentencia FN que va seguida del nombre de la función y el valor de la variable o variables entre paréntesis.

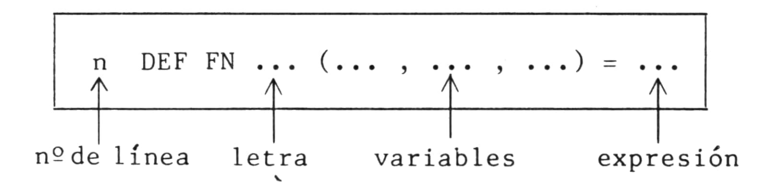
Para la función definida en el ejemplo anterior

```
LET z = FN y(3)
```

asigna a z el valor de la función y para x=3.

PRINT FN y(5) escribirá el valor de la y para x=5.

El formato de la sentencia DEF FN es:



Ejemplo

```
 5  REM PRUEBA DE FUNCION DEFINIDA PARA E=1/2 gt²,
    SIENDO  g=9.8
1Ø  DEF FN E(T) = 1/2*9.8*T↑2
2Ø  FOR T=Ø TO 2Ø STEP 2
3Ø  PRINT T;"..........";FN E(T)
4Ø  NEXT T
5Ø  END
```

La ejecución de este programa da:

```
 Ø .......... Ø
 2 .......... 19.6
 4 .......... 78.4
 6 .......... 176.4
 8 .......... 313.4
1Ø .......... 49Ø
12 .......... 7Ø5.6
14 .......... 96Ø.4
18 .......... 1587.6
2Ø .......... 196Ø
```

Es conveniente tomar la precaución de definir las funciones antes de su uso, puesto que algunos ordenadores así lo exigen.

La mayoría de ordenadores sólo admiten funciones de una variables es decir, funciones del tipo Y(X), Z(T), H(E), etc. ..., en otros se pueden definir funciones de más variables.

Ejemplo:

```
10  DEF FN Z(x,y)=x*x+2*y*y-25
20 ┌FOR x=1 TO 5 STEP 0.5
30 │  ┌FOR y=1 TO 5 STEP 0.5
40 │  │PRINT FN Z(x,y)
50 │  └NEXT y
60 └NEXT x
70  END
```

Este programa calcula unos valores de la función de dos variables: $z(x,y) = x^2+2y^2-25$.

De la misma manera que hemos definido funciones numéricas se pueden definir funciones alfanuméricas, es decir funciones cuyas variables y resultados son cadenas.

El nombre de la función y las variables son de la forma:

A$, X$, Y$, ...

Ejemplo:

El programa:

```
10  DEF FN f$(X$) = X$+"s"
20  INPUT X$
30  PRINT FN f$(X$)
40  END
```

añadirá una s a todas las cadenas que se introduzcan en la línea 20.

En el TEMA X estudiaremos otras funciones que

actúan sobre cadenas.

3. Sentencia ON...GO TO

Cuando la respuesta a una pregunta es sólo si o no, es apropiado usar la sentencia IF...THEN.

Si la ejecución de un programa debe transmitirse a la línea 1000 si x=1, a la 2ØØØ se x=2 y a la 3ØØØ si x=3 (bifurcación múltiple) no sería adecuada la sentencia IF...THEN.

Para solucionar este programa existe la sentencia ON...GO TO que actúa de la siguiente manera:

Si una línea de programa es:

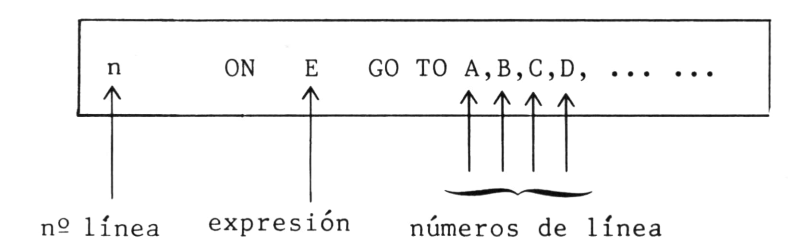
```
n   ON X GO TO 1ØØØ, 21ØØ, 4Ø2Ø
```

al ejecutarla el ordenador hará lo siguiente:

Si INT(x)=1 se transfiere el control o ejecución del programa a la línea 1ØØØ, si INT(x)=2 se transfiere a la línea 21ØØ y si INT(x)=3 se transfiere a la línea 4Ø2Ø. Si x < 1 ó x > 4 ignorará esta línea y la ejecución del programa pasará a la línea siguiente.

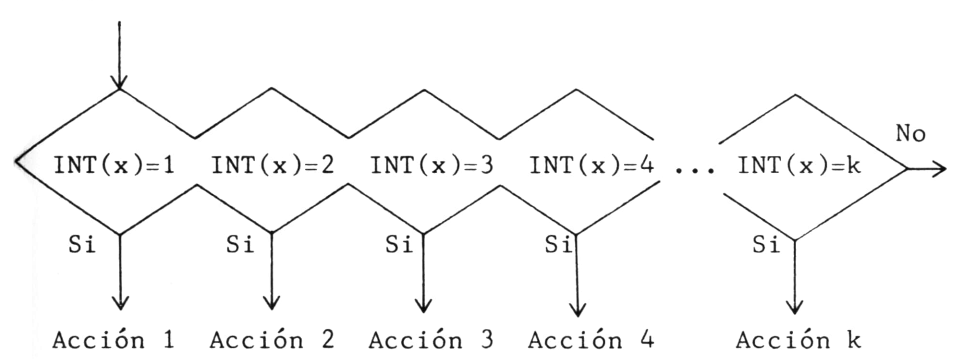
Después del GO TO pueden haber más de 3 números.

El formato de esta sentencia es:



Si INT(E)=1 el control se transfiere a la línea A
Si INT(E)=2 el control se transfiere a la línea B
Si INT(E)=3 el control se transfiere a la línea C
Si INT(E)=4 el control se transfiere a la línea D
....

El diagrama correspondiente a esta bifurcación múltiple es:



Ejemplo

```
  5  REM RECUENTO DE VOTOS DE 1ØØØ VOTANTES
 1Ø  PRINT "Escribe 1 para el voto al candidato A,
     2 para el B y 3 para el C"
 2Ø  LET XA=Ø : LET XB=Ø : LET XC=Ø
 3Ø ┌FOR N=1 TO 1ØØØ
 4Ø │INPUT X
 5Ø │ON X GO TO 1ØØ,2ØØ,3ØØ
 6Ø │PRINT "Voto nulo" : GO TO 31Ø
1ØØ │LET XA = XA+1 : GO TO 31Ø
2ØØ │LET XB = XB+1 : GO TO 31Ø
3ØØ │LET XC = XC+1
31Ø └NEXT N
4ØØ  PRINT "Votos de A  ";XA
41Ø  PRINT "Votos de B  ";XB
42Ø  PRINT "Votos de C  ";XC
43Ø  PRINT "Votos nulos  ";1ØØØ-XA-XB-XC
44Ø  END
```

4. Sentencia ON...GOSUB

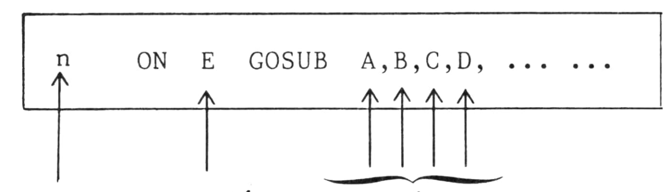
De forma análoga al ON...GO TO existe en BASIC la sentencia ON... GOSUB que actúa de la forma siguiente:

Si la línea n es

n ON X GOSUB 1ØØØ, 2ØØØ, 3ØØØ

entonces si INT(X)=1 se transfiere la ejecución del programa a la subrutina 1ØØØ y regresa a la línea siguiente a la n; si INT(X)=2 se transfiere la ejecución a la subrutina 2ØØØ y regresa a la línea siguiente a la n y si INT(X)=3 se ejecuta la subrutina 3ØØØ y regresa a la línea siguiente a la n. Si x<1 ó x>4 se pasa directamente a la línea siguiente a la n.

El formato es:



EJERCICIOS RESUELTOS

VIII.1. *¿Hay algún error en el siguiente programa?*

```
 1Ø  INPUT a
 2Ø  INPUT b
 3Ø  GO TO 1ØØ
 4Ø  INPUT c
 5Ø  PRINT c*c+a*a+b*b
 6Ø  STOP
1ØØ  PRINT a+b
2ØØ  PRINT a*a+b*b
3ØØ  RETURN
31Ø  END
```

Solución

Hay un error en la línea 3ØØ, la orden es RETURN y no existe antes ninguna orden GOSUB.

* * *

VIII.2. *¿Es correcta la expresión DEF FN 4(x) = 5*x+2 para definir la función f(x) = 5x+2 ?*

Solución

No, pues debería ser una letra en vez del 4, en la definición de la función.

* * *

VIII.3. *Escribir un programa cuya ejecución simule el juego de echar un dado con las siguientes reglas:*

Si sale 3 ó 4 se vuelve a tirar, si sale 1 se pierden 15 ptas., si sale 2 se ganan 6 ptas., si sale 5 ó 6 se ganan 3 ptas. Emplear la sentencia ON...GO TO.

Solución

El programa es:

```
  1Ø  INPUT "Con qué cantidad empiezas el juego?"
      ;G
  2Ø  PRINT "JUEGA PULSANDO CONT"
  3Ø  STOP
  4Ø  LET  D = INT(RND*6)+1
  5Ø  PRINT D
  6Ø  ON D GO TO 1ØØ,2ØØ,3ØØ,3ØØ,4ØØ,4ØØ
 1ØØ  PRINT "Has perdido 15 ptas"
 11Ø  LET G=G-15
 12Ø  PRINT "TOTAL";G
 13Ø  GOSUB 5ØØ
 14Ø  GO TO 2Ø
 2ØØ  PRINT "Has ganado 6 ptas."
 21Ø  LET G=G+6
 22Ø  PRINT " TOTAL";G
 23Ø  GOSUB 5ØØ
 24Ø  GO TO 2Ø
 3ØØ  PRINT "Vuelve a tirar"
 31Ø  PRINT "TOTAL";G
 32Ø  GO TO 2Ø
 4ØØ  PRINT "Has ganado 3 ptas."
 41Ø  LET G=G+3
 42Ø  PRINT "TOTAL";G
 43Ø  GOSUB 5ØØ
 44Ø  GO TO 2Ø
 5ØØ  PRINT "Quieres volver a jugar?(S o N)"
 51Ø  INPUT A$
 52Ø  IF A$="S" THEN RETURN
 53Ø  END
```

* * *

VIII.4. *Hacer un programa para calcular los números combinatorios*

$$\binom{m}{n} = \frac{m!}{(m-n)!n!}$$

usando el programa del factorial como subrutina ($m \geq n$).

Solución

El programa es:

```
 1Ø  INPUT "m=",m
 2Ø  LET k=m
 3Ø  GOSUB 5ØØ
 4Ø  LET a=f
 5Ø  INPUT "n= ",n
 6Ø  LET k=n
 7Ø  GOSUB 5ØØ
 8Ø  LET b=f
 9Ø  LET k=m-n
 95  GOSUB 5ØØ
1ØØ  PRINT a/(b*f)
11Ø  GO TO 56Ø
5ØØ  REM Factorial
51Ø  LET f=1
52Ø ┌FOR i=1 TO k
53Ø │LET f=f*i
54Ø └NEXT i
55Ø  RETURN
56Ø  END
```

* * *

VIII.5. *Escribir un programa en el que introduciendo 1000 letras, imprima las vocales seguidas de s si es a, de n si es e, de r si es i, de p si es o y de m si es u. Emplear unas subrutinas.*

Solución

El programa es:

```
10 ┌FOR n=1 TO 1000
20 │INPUT a$
30 │IF a$="a" THEN GOSUB 100
40 │IF a$="e" THEN GOSUB 200
50 │IF a$="i" THEN GOSUB 300
60 │IF a$="o" THEN GOSUB 400
70 │IF a$="u" THEN GOSUB 500
80 └NEXT n
90  GO TO 510
100 PRINT "as" : RETURN
200 PRINT "en" : RETURN
300 PRINT "ir" : RETURN
400 PRINT "ep" : RETURN
500 PRINT "un" : RETURN
510 END
```

* * *

VIII.6. *Escribir un programa que sirva para calcular los valores de cualquier función, que se introduzca mediante la sentencia DEF FN, en cualquier intervalo y con un incremento de* x *arbitrario.*

Solución

El programa es:

```
10  PRINT "Escribe la función de la siguiente
    forma: 100 DEF FN F(x) = ... (ENTER ó
    RETURN)"
20  PRINT "Luego pulse CONT"
30  STOP
40  INPUT "Intervalo"; a,b
50  INPUT "Incremento de x"; p
60  PRINT "x", "F(x)"
```

```
 7Ø ┌ FOR x=a TO b STEP p
 8Ø │ PRINT x, FN F(x)
 9Ø └ NEXT x
11Ø   END
```

* * *

VIII.7. *Escribir un programa en el que se defina la función*

$$f(x) = \frac{x - 1}{\cos x}$$

y calcule: $A = f(a)+f^3(b)$ *introduciendo de antemano a y b.*

Solución

El programa es:

```
10  INPUT a,b
20  DEF FN F(x) = (x-1)/COS(x)
30  PRINT "A=";FN F(a) + FN F(b) * FN F(b) *
    FN F(b)
40  END
```

* * *

EJERCICIOS PROPUESTOS

275. Tanto la instrucción GO TO como GOSUB se utilizan para efectuar un salto en el programa. ¿Alguna de ellas se complementa con otra instrucción?

276. En un programa ¿debe haber el mismo número de sentencias GOSUB que de RETURN?

*277. ¿Se puede enviar el control de un programa a una subrutina mediante un GO TO?

278. ¿Hay algún error en el siguiente programa?

```
 1Ø  INPUT a
 2Ø  IF a > 3  THEN GO TO 1ØØ
 3Ø  PRINT b
 4Ø  GO TO 13Ø
1ØØ  REM Subrutina
11Ø  LET b=a+3
12Ø  RETURN
```

279. ¿Qué ocurriría con la ejecución de un programa que contuviera las siguientes líneas?

```
...  ...  ...
4Ø  GOSUB 1ØØ
...  ...  ...
1ØØ  REM Subrutina
...  ...  ...
...  ...  ...
16Ø  GOSUB 5ØØ
...  ...  ...
```

```
...  ...  ...
200  RETURN
...  ...  ...
...  ...  ...
500  REM Subrutina
...  ...  ...
...  ...  ...
540  GOSUB 100
...  ...  ...
...  ...  ...
560  RETURN
...  ...  ...
```

280. ¿Es correcto un programa que contenga las siguiente líneas?

```
100  GOSUB 300
...  ...  ...
...  ...  ...
300  REM Subrutina
...  ...  ...
...  ...  ...
400  RETURN 100
...  ...  ...
...  ...  ...
```

*281. ¿Cuál es el resultado de la ejecución del siguiente programa?

```
10 ┌FOR n=1 TO 4
20 │READ x
30 │GOSUB 70
40 └NEXT n
50  STOP
60  DATA 2,5,8,9
70  PRINT "La superficie del cuadrado
    de lado";x;"es";x↑2
80  RETURN
```

*282. ¿Es correcto el programa

```
10  DEF FN B(x) = 3*FN A(x)
20  DEF FN A(x) = 2*x
30  PRINT 4, FN A(4), FN B(4)
```

en el que se definen las funciones A(x) y B(x)? Nótese que B(x) la definimos antes que A(x), ¿tiene alguna importancia esto en el programa?

283. ¿Es correcta la expresión:

```
DEF FN A(x) = 3* FN A(x) ?
```

*284. Escribir un programa que nos dé el valor de

$$1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$$

para n=15 utilizando una subrutina para el cálculo del factorial.

Comparar el resultado del mismo con el número e (e=2,718281828...)

285. Escribir un programa en el que desde n=3 hasta 100 calcule el área del polígono regular de n lados inscrito en una circunferencia de radio r (introducido de antemano) si n es par y el área del polígono regular circunscrito si n es impar. Emplear unas subrutinas.

286. Escribir un programa en el que introduciendo 100 ángulos en grados, minutos y segundos si son menores que 45º imprima las razones trigonométricas seno y coseno y si son mayores o iguales que 45º imprima la tantenge y cotangente. Emplear una subrutina.

287. En un programa se pide al usuario que introduzca un número positivo. Escribir una subrutina que en el caso de ser negativo le pida volver a introducir el dato correcto y si es positivo asigne a la variable x la parte entera de este número.

288. Escribir una subrutina que empieze en la instrucción 5ØØ y que de tres variables a,b,c elija la menor asignando su valor a la variable m pero quedando inalterados los valores de a,b,c.

289. Escribir una parte de un programa en el que se introduzca una variable x. Si x=1 envíe la ejecución del programa a la subrutina 1ØØØ, si x=2 a la 2ØØØ y si x=3 a la 3ØØØ.

290. Escribir un programa empleando la sentencia ON...GO TO... que informe sobre:

a) Número de tomos de una biblioteca.
b) Número de tomos de Historia.
c) Número de tomos de Ciencias.
d) Número de revistas.

previamente almacenados con DATA.

291. Escribir un programa en el que se defina la función

$$f(x) = (x^2+4x+1)e^x$$

y calcule: f(a)+f(b) para el siguiente conjunto de valores de a y b:

$$\begin{cases} a=0 \\ b=1,5 \end{cases}, \quad \begin{cases} a=2 \\ b=-3 \end{cases}, \quad \begin{cases} a=-5,1 \\ b=1/3 \end{cases}, \quad \begin{cases} a=3/5 \\ b=-1/4 \end{cases}$$

292. Escribir un programa en el que se definan las funciones:

$$Sh(x) = \frac{e^x - e^{-x}}{2} \qquad \text{(seno hiperbólico)}$$

$$Ch(x) = \frac{e^x + e^{-x}}{2} \qquad \text{(coseno hiperbólico)}$$

y que calcule a) $Sh(3) + Ch^2(5)$
b) $Sh^3(1/2)+Ch(4,3)$

*293. Escribir un programa que defina la función:

$$f(a,b,c) = 2a + 3b + 4c + 2ab + 3bc$$

y calcule

$$z = f(a,b,c) + f^2(a,b,c) + f^3(a,b,c)$$

para los números a,b,c enteros, tales que

$$0 \leqslant a \leqslant 7 \quad , \quad 3 \leqslant b \leqslant 14 \quad , \quad 5 \leqslant c \leqslant 12$$

294. Escribir un programa para calcular valores de la función

$$f(x) = \begin{cases} 3x-1 & \text{si} \quad x \leqslant 0 \\ x^2 & \text{si} \quad x > 0 \end{cases}$$

entrando el valor de x mediante la sentencia INPUT y utilizando la función definida.

*295. Escribir una subrutina para calcular los valores de $(g \circ f)(x)$ siendo $f(x)$ y $g(x)$ funciones ya definidas en un programa.

296. Escribir un programa que imprima la letra de una canción cuyo estribillo se repita varias veces

empleando una subrutina para el mismo.

297. Escribir un programa que imprima un soneto de Lope de Vega de manera que despues de cada cuarteto y de cada terceto imprima el gráfico

* * *

298. Escribir un programa que para dos números complejos

$$\begin{cases} z_1 = a+bi \\ z_2 = c+di \neq 0 \end{cases}$$

el usuario pueda escoger las siguientes opciones (MENU):

1 Calcular z_1+z_2
2 Calcular $z_1 \cdot z_2$
3 Calcular z_1/z_2

escrita cada una de las opciones como una subrutina.

299. Escribir un programa que para un polinomio cualquiera de 2º grado $p(x) = ax^2+bx+c$ el usuario pueda escoger las siguientes opciones (MENU):

1 Calcular el valor numérico de $P(x)$.
2 Resolver la ecuación: $P(x) = 0$
3 Calcular el vértice de la parábola
$y = P(x) \quad (x = \frac{-b}{2a}, \quad y = \frac{4ac-b^2}{4a})$

escrita cada una de las opciones como una subrutina.

9
VARIABLES CON SUBINDICE. SENTENCIA DIM

Recordemos que en BASIC existen dos tipos de variables:

a) **Numéricas**, representadas por una o más letras o por una letra seguida de números o letras.

b) **Alfanuméricas** o variables de cadenas, representadas como las numéricas seguidas del signo $ (Algunos ordenadores sólo admiten una letra seguida de $).

Con este tipo de representación si hay un número elevado de variables es engorroso ponerles letras distintas a cada una.

Es de uso común que los ordenadores puedan tratar cantidades ingentes de datos, para ello se utilizan las variables con subíndice en forma de listas (vectores) y tablas (matrices).

1. Listas (vectores)

Supongamos que en un programa se necesita manejar la edad de los 600 alumnos de un Instituto. Esto puede hacerse considerando las edades como elementos de una lista, representando las edades por un nombre

(una letra) seguido de un subíndice encerrado con un paréntesis. O sea se pueden definir las variables E(1), E(2),...,E(600), siendo:

E(1) la edad del primer alumno.
E(2) la edad del segundo alumno.
....
E(6ØØ) la edad del seiscientosavo alumno.

Así pues los conjuntos de variables con un subíndice o listas se representan por una variable (numérica o alfanumérica) seguida de un número entre parentesis (que es el subíndice); todas las variables tienen el mismo nombre y sólo se distinguen por el subíndice.

Ejemplo:
p(1) = 2
p(2) = 4
p(3) = 6
p(4) = 8

sería una lista de 4 variables que tomarían los valores

2 , 4 , 6 , 8

Estas variables con subíndice se utilizan igual que las variables simples. Por ejemplo si en una línea de programa se encuentra:

```
n  LET x = 3 * p(4)
```

entonces x tomaría el valor 24 (3 * 8)

Si se encuentra:

```
n  PRINT p(3)
```

escribirá 6.

Si se desea almacenar en un programa 200 nombres podemos usar variables de cadena con subíndices. Estas se representan con una variable alfanumérica seguida de un número entre paréntesis que es el subíndice.

Así

n$(1)   es el primer nombre
n$(2)   es el segundo nombre
....
n$(200) es el doscientosavo nombre

Si una línea de programa es:

n PRINT n$(45)

escribirá el nombre que ocupa el lugar 45 en la lista n$(i).

2. Tablas (matrices)

Pueden utilizarse también variables numéricas o alfanuméricas con dos subíndices (tablas). Cada elemento de la tabla queda determinado por una variable numérica o alfanumérica seguida de dos números entre paréntesis separados por una coma. El primero de los subíndices indica la línea y el segundo la columna en la tabla.

Estas variables son pues del tipo

A(2,5), M(3,10), p$(4,7), ...

A(2,5) representa el valor numérico de la fila 2, columna 5 de la tabla A.

M(3,1Ø) representa el valor numérico de la fila 3, columna 10 de la tabla M.

p$(4,7) representa la cadena de la fila 4, columna 7 de la tabla de cadenas p$.

Ejemplo

Si la tabla R es:

$$\begin{pmatrix} 1 & 4 & 6 & 7 \\ 2 & -3 & 5 & 3 \\ -5 & 6 & 8 & 10 \end{pmatrix}$$

entonces

$$\begin{cases} R(2,3) & \text{toma el valor } 5. \\ R(3,4) & \text{toma el valor } 10. \end{cases}$$

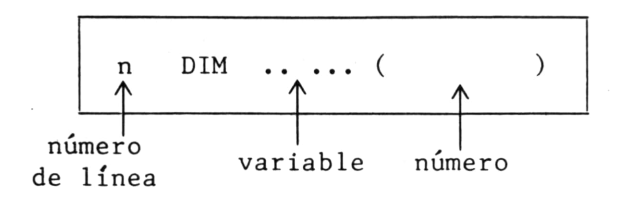
3. Sentencia DIM (dimensión)

La sentencia DIM se utiliza para que el ordenador guarde la memoria suficiente para almacenar variables con subíndice (listas y tablas).

Suele escribirse al principio del programa asegurándonos el suficiente espacio de memoria para las listas y tablas que se necesitan.



El formato de la sentencia DIM para las listas es como sigue:



Ejemplo

Si una línea de programa es:

n DIM A(2Ø)

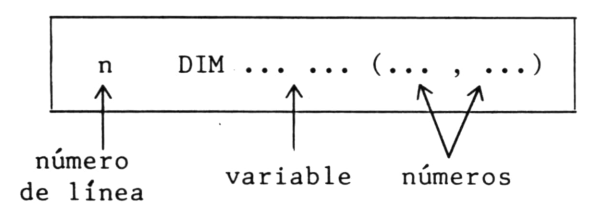
el ordenador reserva en la memoria espacio suficiente para una lista de nombre A con 20 variables A(1),A(2),...,A(20).

Si es:

n DIM B$(3Ø)

el ordenador reserva espacio para 30 cadenas de nombre B.

El formato de la sentencia DIM para tablas es:



Ejemplo

Si una línea de programa es

n DIM A(4,6)

reserva espacio para una tabla numérica de 4 filas y 6 columnas.

Si es:

n DIM A$(5,8)

reserva espacio para una tabla de cadenas de 5 filas y 8 columnas.

4. Algunas cuestiones prácticas

- Para utilizar las variables con subíndice se pueden nombrar en forma explícita

A(3)
B(2,4)
C$(8)
...

pero también es válido utilizar expresiones del tipo:

A(m), B(2*x), C$(k-1), ...

donde el subíndice está en función de las variables m,x,k,...

- También es válido utilizar expresiones DIM del tipo

n DIM A(k)

o sea que el número de elementos de la lista A depende del valor que tome la variable k (la cual naturalmente debe estar definida antes de la sentencia DIM).

- Algunos ordenadores admiten el sunbíndice Ø y otros deben empezar en 1. Si utilizamos como norma subíndices que empiecen por 1 tendremos más seguridad de que nuestro programa corra bien en el cualquier ordenador.

- No es recomendable la misma letra para una variable con subíndice y para una variable simple.

- Para asignar valores a las variables con subíndices suele utilizarse las sentencias DATA, READ si cada vez que se ejecuta el programa deben tomar los mismos valores y la sentencia INPUT si es necesario que cambien en cada ejecución del programa.

Ejemplo 1

```
10  DIM A(4)
20  DATA 2,24,30,50
30 ┌FOR I=1 TO 4
40 │READ A(I)
50 └NEXT I
..................
```

Esta parte de programa asignará los valores

A(1) = 2,   A(2) = 24,   A(3) = 30,   A(4) = 50

Ejemplo 2

```
10  DIM A(4)
20 ┌FOR I=1 TO 4
30 │INPUT A(I)
40 └NEXT I
```

Esta parte de programa permitirá al usuario introducir los valores de A(1), A(2), A(3) y A(4) que convenga en cada caso.

Vamos a escribir ahora un programa que permita calcular la media de una cantidad de números cada vez distinta sin destruir estos números o datos, de forma que en el mismo programa estos pudieran utilizarse para otros cálculos (por ejemplo la desviación típica).

```
10  INPUT "Número de datos";k
20  DIM A(k)
30  LET S=0
40 ┌FOR I=1 TO k
50 │INPUT A(I)
60 │LET S=S+A(I)
70 └NEXT I
80  LET M=S/k
90  PRINT "Media   =   ";M
```

Nótese que el programa podría continuar utilizando los datos A(1),A(2),...,A(k) para otros cálculos.

· Notas

1. En algunos ordenadores no es necesario dimensionar las listas o tablas con menos de 11 elementos, o sea que admitirían variables del tipo N(10) o bien N(10,10) sin haberlos dimensionado antes.

2. En algún ordenador al escribir DIM A$(20,12) significa una lista de 20 cadenas de 12 caracteres cada una (entendiendo que si hay menos de 12 el resto son espacio en blanco). Esto es equivalente a una tabla de 20 filas y 12 columnas con un caracter único en cada lugar de la tabla.

EJERCICIOS RESUELTOS

IX.1. *¿Dará mensaje de error la ejecución del siguiente programa?*

```
10  DIM N(11)
20 ┌FOR i=1 TO 12
30 │READ N(i)
40 │PRINT N(i)
50 └NEXT i
60  DATA 1,2,3,4,5,6,7,8,9,10,11,12
70  END
```

Solución

Sí que dará mensaje de error puesto que en la línea 10 sólo se dimensionan 11 variables con subíndice y se utilizan 12.

* * *

IX.2. *Hacer un programa en el que el ordenador pida una tabla de valores de 12 filas y 8 columnas entrándola por filas y calcule la suma de las columnas.*

Solución

El programa es:

```
10  DIM A(12,8)
20  PRINT "Introducir los valores por filas"
30 ┌FOR I=1 TO 12
40 │   ┌FOR J=1 TO 8
50 │   │INPUT A(I,J)
60 │   └NEXT J
70 └NEXT I
80  CLS
```

```
 9Ø ┌FOR J=1 TO 8
1ØØ │LET S=Ø
11Ø │  ┌FOR I=1 TO 12
12Ø │  │LET S=S+A(I,J)
13Ø │  └NEXT I
14Ø │   PRINT "La suma de la columna   ";J;
    │   "  es  ";S
15Ø └NEXT J
16Ø  END
```

* * *

IX.3. *Las cuatro notas de 14 alumnos son:*

Alumno	Notas
1	4,2,5,2
2	1,7,10,4
3	8,4,5,1
4	6,6,7,6
5	7,9,9,7
6	8,4,6,4
7	5,2,6,1
8	4,6,5,8
9	5,4,6,7
10	6,1,1,2
11	4,4,5,6
12	3,6,6,1
13	2,3,5,5
14	7,1,7,4

Escribir un programa que dé la nota media de cada uno, el número de lista que le corresponde y que extraiga de esta lista aquellos cuya nota media es mayor o igual que 5.

Solución

El programa es:

```
10  LET T=0
20  DIM N(14,4): DIM S(14)
30  PRINT "Las notas medias son:"
40 ┌FOR i=1 TO 14
50 │   ┌FOR J=1 TO 4
60 │   │READ N(i,j)
70 │   │LET T=T+N(i,j)
80 │   └NEXT J
90 │ PRINT "Alumno  ";i;"...";T/4
100│ LET S(i) = T/4
110│ LET T=0
120└NEXT i
130 PRINT : PRINT
140 PRINT "Los alumnos cuya nota media es
    mayor o igual que 5 son:"
150 PRINT : PRINT
160┌FOR i=1 TO 14
170│IF S(i)>= 5 THEN PRINT i;"...";S(i)
180└NEXT i
190 DATA 4,2,5,2,1,7,10,4,... (hasta los
    14 alumnos)
200 END
```

* * *

IX.4. *Escribir un programa para almacenar el recuento de votos de la siguiente tabla y que imprima el número de votantes menores de 25 años, el de los que tienen 25 años o más y el número de votos de cada partido.*

		PARTIDOS			
		A	B	C	D
NUMERO DE VOTANTES	Menores de 25 años.	54	28	35	47
	De 25 años o más.	68	53	61	60

Solución

El programa es:

```
10   DIM A$(6) : DIM N(2,4)
20  ┌FOR I=1 TO 6
30  │READ A$(I)
40  └NEXT I
50  ┌FOR I=1 TO 2
60  │  ┌FOR J=1 TO 4
70  │  │READ N(I,J)
80  │  └NEXT J
90  └NEXT I
120 ┌FOR I=1 TO 2
125 │LET F=0
130 │   ┌FOR J=1 TO 4
140 │   │LET F=F+N(I,J)
150 │   └NEXT J
160 │PRINT "Votantes  ";A$(I+4);TAB(29);F
180 └NEXT I
190 ┌FOR J=1 TO 4
195 │LET C=0
200 │  ┌FOR I=1 TO 2
210 │  │LET C=C+N(I,J)
220 │  └NEXT I
230 │PRINT "Votantes del partido  ";A$(J);
    │TAB(29);C
240 └NEXT J
250  DATA "A","B","C","D","Menores de 25 años",
     "De 25 años o más"
260  DATA 54,28,35,47,68,53,61,60
270  END
```

* * *

IX.5. *Escribir un programa cuya ejecución ordene de menor a mayor una lista de k números.*

Solución

El programa es:

```
10   DIM N(k)
20 ┌FOR I=1 TO k
30 │READ N(I)
40 └NEXT I
50 ┌FOR A=1 TO k-1
60 │  ┌FOR B=1 TO k-1
70 │  │LET X=N(B)
80 │  │LET Y=N(B+1)
90 │  │IF X<=Y THEN GO TO 200
100│  │LET N(B)=Y
110│  │LET N(B+1)=X
200│  └NEXT B
210└NEXT A
220┌FOR C=1 TO k
230│PRINT N(C)
240└NEXT C
250  DATA (lista de números)
260  END
```

Nota

Esta forma de ordenar se llama **método de la burbuja.** Se podrían entrar los números N(I) mediante la sentencia INPUT.

Otro método:

```
10   INPUT "Cuantos números hay que ordenar?"
     ;k
20   DIM N(k)
30 ┌FOR I=1 TO k
40 │INPUT N(I)
50 └NEXT I
60 ┌FOR I=2 TO k
70 │  ┌FOR J=1 TO I-1
80 │  │IF N(I)< N(J)  THEN LET X=N(J):
   │  │LET N(J)=N(I) : LET N(I)=X
90 │  └NEXT J
100└NEXT
110  CLS
```

```
120 ┌ FOR I=1 TO k
130 │ PRINT N(I)
140 └ NEXT I
150   END
```

Nota

Estos programas también sirven para ordenar cadenas cambiando N por N$.

* * *

IX.6. *En una tienda se venden los siguientes artículos:*

1. *Champú de 120 ptas. unidad*
2. *Jabón líquido a 110 ptas. unidad*
3. *Colonia a 200 ptas. unidad.*
4. *Suavizante a 145 ptas. unidad.*
5. *Detergente a 638 ptas. unidad.*
6. *Lejía a 89 ptas. unidad.*
7. *Desodorante a 180 ptas. unidad*
8. *Jabón de afeitar a 248 ptas. unidad.*
9. *Esponjas a 80 ptas, unidad.*
10. *Gel a 215 ptas. unidad.*

Escribir un programa en el que introduciendo el número del artículo y el número de unidades compradas por cada cliente imprima la factura de dicha compra detallando: el nombre del artículo, el número de unidades, el precio por unidad, el total de cada artículo y el total de la compra.

Solución

El programa es:

```
10  DIM A$(10) : DIM p(10) : DIM r(10) : DIM u(10)
20 ┌FOR i=1 TO 10
30 │READ A$(i)
40 │READ p(i)
50 └NEXT i
60 ┌FOR i=1 TO 10
70 │INPUT "Artículo (20 si no hay más)";r(i)
80 │IF r(i)=20 THEN GO TO 110
90 │INPUT "Unidades";u(i)
100└NEXT i
105 LET n=10 : LET t=0 : GO TO 120
110 LET n=i-1 : LET t=0
120 PRINT "Art.";TAB(12);"N.uni.";TAB(20);"P.uni."
    ;TAB(27);"Total"
125 PRINT "---";TAB(12);"------";TAB(20);"-----";
    TAB(27);"-----"
130┌FOR i=1 TO n
140│LET P=u(i)*p(r(i))
150│PRINT : PRINT A$(r(i));TAB(15);u(i);TAB(20);
   │p(r(i));TAB(27);P
160│LET t=t+P
170└NEXT i
175 PRINT "-----------------------------------"
180 PRINT : PRINT : PRINT
190 PRINT TAB(7);"Total compra:----";TAB(28);t
195 PRINT : PRINT : PRINT "Gracias por su visita"
200 DATA "Champú",120,"Jabón líquido",110,"Coloni
    a",200,"Suavizante",145,"Detergente",638,"Lej
    ia",89,"Desodorante",180 "Jabón afeitar",248,
    "Esponjas",80,"Gel",215
210 END
```

EJERCICIOS PROPUESTOS

300. ¿Cuáles de las siguientes variables son variables con subíndice ?

N1, A(2), N(12), M4, B6

301. ¿De qué clase son las siguientes variables?
   a) ASB(3,5)
   b) PAN$
   c) CADENA
   d) P
   e) Pi(5)
   f) AR$(3,5)
   g) COT$(5)

*302. ¿Qué imprimirá la ejecución del siguiente programa?

```
 1Ø  DIM A(15)
 2Ø ┌FOR I=1 TO 15
 3Ø │READ A(I)
 4Ø └NEXT I
 5Ø  LET S = Ø
 6Ø ┌FOR J=1 TO 15
 7Ø │LET S=S+A(J)
 8Ø │PRINT A(J)
 9Ø └NEXT J
1ØØ  PRINT "M.A.=";S/15
11Ø  DATA 2.3,4.5,Ø.5,3.8,4.2,6.7,7.3,4.9,5.Ø,1.2,
     4.5,6.5,4.5,4.3,1.7
12Ø  END
```

303. ¿Qué imprimirá la ejecución del siguiente programa?

```
1Ø  DIM N(6,4)
2Ø ┌FOR i=1 TO 6
3Ø │  ┌FOR j=1 TO 4
4Ø │  │READ N(i,j)
5Ø │  │IF i<J THEN PRINT N(i,j)
6Ø │  └NEXT j
7Ø └NEXT i
8Ø  DATA 1,2,3,4,5,6,7,8,9,1Ø,11,12,13,14,15,16,
    17,18,19,2Ø,21,22,23,24
9Ø  END
```

304. ¿Qué imprimirá la ejecución del siguiente programa ?

```
 1Ø  DATA 1,2,3,4,5,6,7,8,9,1Ø,11,12,13,14
 2Ø  DIM A(12) : DIM B(7)
 3Ø ┌FOR i=1 TO 12
 4Ø │READ A(i)
 5Ø └NEXT i
 6Ø  RESTORE
 7Ø ┌FOR k=1 TO 7
 8Ø │READ B(k)
 9Ø └NEXT k
1ØØ  PRINT A(4) : PRINT B(4)
11Ø  END
```

305. Escribir un programa cuya ejecución almacene mediante variables con subíndice los meses del año.

* 306. Hacer un programa que pida dos vectores A y B de $\mathbb{R}^3$ y calcule el producto escalar de ambos y las componentes del producto vectorial.
(Producto escalar:

$$(a_1,a_2,a_3)\cdot(b_1,b_2,b_3) = a_1b_1 + a_2b_2 + a_3b_3$$

Producto vectorial:

$$(a_1,a_2,a_3)\wedge(b_1,b_2,b_3) = (a_2b_3-a_3b_2,a_3b_1-a_1b_3,a_1b_2-a_2b_1)$$

307. Hacer un programa para resolver sistemas lineales de dos ecuaciones con dos incógnitas.

308. Hacer un programa en el que introduciendo dos tablas o matrices A y B de 5 filas por 4 columnas cada una, imprima la tabla o matriz que se obtiene sumando los elementos correspondientes.

309. Hacer un programa en el que introduciendo N números mediante INPUT calcule la media, la varianza y la desviación típica.

$$\text{Media} = \bar{x} = \frac{\sum_{i=1}^{N} x_i}{N}$$

$$\text{Varianza} = \sigma^2 = \frac{\sum_{i=1}^{N} (x_i - \bar{x})^2}{N}$$

$$\text{desviación típica} = \sqrt{\sigma^2} = \sigma$$

310. Hacer un programa para que el ordenador pida al usuario una tabla de valores enteros de 15 filas por 6 columnas e imprima estos valores en forma de tabla de 15 filas por 6 columnas.

*311. Escribir un programa en el que introduciendo 40 números en una tabla de 8 filas por 5 columnas los escriba en forma de tabla y luego vuelva a imprimir la tabla en orden creciente para cada fila.

312. Una tienda de venta al por mayor tiene cada semana 6 artículos de oferta de la siguiente manera: Hasta 5 unidades de cualquier artículo de oferta descuenta el 2%, de 6 a 10 unidades el 4%, de 10 a 15 unidades el 6% y más de 15 unidades el 10%. Hacer un programa en el que introduciendo el precio de los 6 artículos en una línea DATA nos dé como resultado la tabla de los precios por unidad según el número de unidades que se compran respecto de cada artículo.

O sea:

Si n es el número de unidades

	$n \leqslant 5$	$5 < n \leqslant 10$	$10 < n \leqslant 15$	$n > 15$
art.1				
art.2				
art.3				
art.4				
art.5				
art.6				

313. Escribir un programa para ordenar en orden alfabético N nombres introducidos mediante un DATA.

*314. Un profesor ha puesto dos exámenes a un grupo de 40 alumnos y quiere tener almacenadas las notas y la nota media de los dos exámenes para cada alumno. Hacer el programa correspondiente.

315. Añadir lo que convenga al programa del problema anterior para que además de la nota media el ordenador imprima: MD, I, S, B, N, E según el siguiente baremo: (M = nota media)

$M < 2,5$ es MD (Muy deficiente)
$2,5 \leq M < 5$ es I (Insuficiente)
$5 \leq M < 6$ es S (Suficiente)
$6 \leq M < 7$ es B (Bien)
$7 \leq M \leq 8$ es N (Notable)
$8 < M$ es E (Excelente)

316. Hacer un programa que almacene las notas media de 30 alumnos en una variable con subíndice y dibuj en diagrama horizontal mediante signos "" poniend uno por cada alumno que tenga MD,I,S,B,N,E segú el baremo del problema anterior.

317. Escribir un programa cuya ejecución almacen las notas de 10 alumnos en las 5 asignaturas de Matemá ticas (1), Informática (2), Ciencias (3), Física (4) Química (5) y obtenga para cada asignatura la not media.

Aplicarlo al caso:

1 Mat.	5	4	4	6	3	7	2	7	8	5
2 Inf.	4	5	5	3	5	6	7	5	6	8
3 Cienc.	7	8	6	5	4	4	5	5	5	6
4 Fís.	4	3	1	6	4	5	4	6	9	7
5 Quím.	4	5	2	6	5	4	5	7	7	6

318. Una empresa fabrica 4 artículos y los vend a través de 12 distribuidores. Hacer un programa que utilizando una variable con dos subíndices, almacen el número de ventas de cada artículo para cada distri buidor y calcule el precio total de los artículo vendidos por cada distribuidor formando con ello una lista, calculando también el número total de ventas (Utilizar una línea DATA para los precios de cad artículo).

319. Hacer un programa que introduzca una tabla A de 3 filas y 3 columnas y una lista de 3 elementos y calcule la suma de los productos de los elementos de cada fila de A por los elementos respectivos de B guardando los resultados en una lista C imprimiendo esta lista.

320. Hacer un programa en el que introduciendo 12 nombres de un grupo de personas con las edades de cada uno de ellos nos dé como resultado los nombres con la edad y además la edad media del grupo.

321. Escribir un programa en el que introduciendo con la sentencia INPUT los resultado de una encuesta sobre 6 conjuntos musicales: Rock, Ritmo, Vals, Melodía, Movida y Saltos, nos dé como resultado el número de votos de cada conjunto y los votos nulos.

*322. Escribir un programa en el que dadas 5 letras cualesquiera dé como resultado todas las formas distintas de escribir las mismas (formar las permutaciones de 5 elementos).

10
FUNCIONES DE CADENAS

En este tema estudiaremos funciones cuyo argumento es una **cadena.** Son por tanto funciones que actúan sobre las cadenas y cuyo resultado es otra cadena o un número.

O sea:

	Función de cadena	{Cadenas}
{Cadenas}	⟶	ó
		{Números reales} (ℝ)

1. Función LEN (longitud)

Esta función tiene como argumento una cadena cualquiera (A$) y actúa de la siguiente manera:

LEN(A$) da el número de caracteres de A$

Notas

a) No hace falta colocar la cadena argumento en forma de variable alfanumérica. Se puede colocar entre co-

millas directamente.

b) Recuérdese que los espacios en blanco cuentan como caracteres.

Ejemplos

a) Si A$ = "ALMA", entonces PRINT LEN (A$) da como resultado 4.

b) PRINT LEN ("BARCELONA") da 9

c) PRINT LEN ("MES") + LEN("DE") + LEN("ENERO") da 1Ø

d) PRINT LEN ("MES DE ENERO") da 12

e) El programa siguiente:

```
10  LET A$="LA PALABRA  "
20  INPUT B$
30  LET N=LEN(B$)
40  PRINT A$;B$;"  TIENE  ";N;"  CARACTERES"
```

imprimirá el número de caracteres de B$. Por ejemplo si B$ = "PERRO" imprimirá:

LA PALABRA PERRO TIENE 5 CARACTERES

2. Subcadenas

Entendemos por subcadena de A$ a una cadena cuyos elementos son de A$ conservando el orden.

Ejemplos

a) "BAR","ONA","CEL","BRCN" son subcadenas de "BARCELONA".

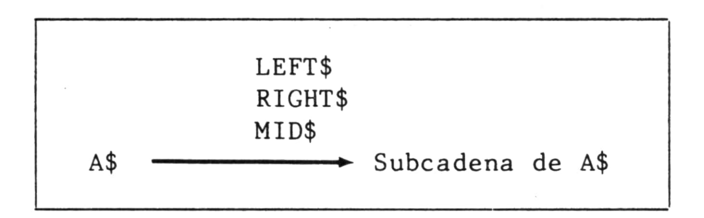
b) "PARIS CAPITAL" es una subcadena de "PARIS ES LA CAPITAL DE FRANCIA"

3. Extracción de subcadenas de una cadena dada

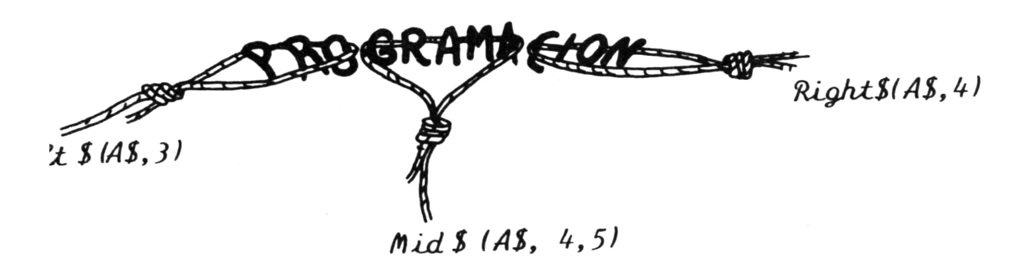
Existen tres funciones de cadena cuyo resultado es una subcadena:

- LEFT$
- RIGHT$
- MID$

o sea:



A$ = "Programación"



que actúan de la siguiente manera:

LEFT$(A$,N)	da la subcadena de A$ formada por los N primeros caracteres de A$.
RIGHT$(A$,N)	da la subcadena de A$ formada por los N últimos caracteres de A$.
MID$(A$,P,N)	da la subcadena de A$ formada por los N caracteres de A$ a partir del caracter que ocupa el lugar P (éste incluído).

Como, por regla general, una cadena puede tener a lo sumo 255 caracteres, debe ser:

$$0 < N \leqslant 255$$

Notas

a) LEFT significa **izquierda** en inglés.
RIGHT significa **derecha** en inglés.
MID es abreviación de la palabra inglesa **middle** que significa **medio.**

b) No hace falta colocar la cadena argumento en forma de variable alfanumérica. Se puede colocar entre comillas directamente.

c) Los espacios en blanco cuentan como caracteres.

Ejemplos

a) Si A$ = "ALMA" entonces
PRINT LEFT$(A$,2) da AL
PRINT RIGHT$(A$,3) da LMA
PRINT MID$(A$,2,1) da L

b) PRINT LEFT$ ("SEVILLA",3) da SEV
PRINT RIGHT$ ("SEVILLA",4) da ILLA
PRINT MID$ ("SEVILLA",3,4) da VILL

c) PRINT LEFT$("ARTICULO",2) + MID$("ABORIGEN",2,2) + RIGHT$("PANEL",1) da ARBOL

d) PRINT LEFT$("EL EQUIPO LOCAL",9) da EL EQUIPO
PRINT MID$("EL EQUIPO LOCAL",4,6) da EQUIPO
PRINT RIGHT$("EL EQUIPO LOCAL",12) da EQUIPO LOCAL

e) El programa siguientee:

```
10 INPUT A$
20 PRINT TAB(4); LEFT$(A$,12)
30 PRINT
40 PRINT TAB(8); MID$(A$,14,5)
50 PRINT
60 PRINT TAB(6); RIGHT$(A$,8)
70 END
```

si A$ = "EL TREN SALE A LAS 15 HORAS", imprimirá:



4. Función VAL (valor)

La función VAL tiene como argumento una cadena A$ y actúa de la siguiente manera:

VAL(A$)	da el número formado por las cifras que hay en la cadena A$ a partir del principio hasta que encuentra un caracter no numérico distinto del espacio vacío.

Ejemplos

a) PRINT VAL("153 de diagonal") da 153

b) PRINT VAL("La diagonal es 153") da Ø

c) PRINT VAL("1 2 3) da 123

d) PRINT VAL (1.45 de altura) da 1.45

Existe en algunos dialectos del BASIC una variante de la función VAL que convierte la cadena argumento A$ en su valor numérico si A$ expresa un cálculo.

O sea:

> VAL(A$) da el valor de A$
> (A$ expresa un cálculo)

Ejemplos

a) Si A$ = "SQR(4)" VAL(A$) da 2

b) PRINT VAL("COS(π/3)")da Ø.5

c) PRINT VAL("3") da 3

d) El programa

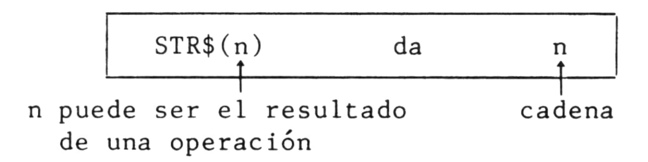
```
10  INPUT A$
20  LET x=4
30  PRINT VAL(A$)
40  END
```

si A$ = "(x+1)(x-3)", imprimirá: 5

5. Función STR$(cadena)

La función STR$ convierte un número en una cadena.

O sea:



Ejemplos

a) PRINT STR$(3) da 3

b) PRINT STR$(VAL("12AB")) da 12

c) PRINT STR$(12) + STR$(23) da 1223

d) PRINT VAL(STR$(3*5)) da 15

e) El programa:

```
10  LET N$ = STR$(3 ↑ 2)
20  PRINT N$
30  END
```

imprimirá: 9.

6. Código

En el lenguaje BASIC a cada carácter (letra, número, signo, espacio en blanco) se le asocia un número que es su **código** llamado ASCII (América National Standar Code for Information Interchange).

La relación de este código está al final de este Tema.

Existen dos funciones para relacionar caracteres con códigos.

La función ASC aplicada a una cadenas A$ da el código del primer carácter de A$.

Nota

El código ASCII está relacionado con el orden de las cadenas pues el ordenador las compara por el número de código de su primer carácter. Si éste es igual por el del segundo carácter y así sucesivamente.

Esto completa la ordenación de cadenas explicada en el Tema V.

Ejemplos

a) PRINT ASC("A") da 65

b) Si A$ = "49"
PRINT ASC(A$) da 52 que es el código de "4".

c) PRINT ASC("B")+ASC("C") da 133

7. Función CHR$ (carácter)

La función CHR$ se aplica a números comprendidos entre 32 y 126 y actúa de la siguiente manera:

CHR$(n) da el carácter de código n

Ejemplos

a) PRINT CHR$(65) da A

b) PRINT CHR$(49) da 1

c) El siguiente programa muestra el código ASCII.

```
1Ø   PRINT TAB(1Ø);"CODIGO ASCII"
2Ø   PRINT TAB(1Ø);"------------"
3Ø ┌ FOR  I=32  TO 126
4Ø │ PRINT I, CHR$(I)
5Ø └ NEXT I
6Ø   END
```

imprimirá:

Código	Carácter	Código	Carácter	Código	Carácter
32	espacio	52	4	72	H
33	!	53	5	73	I
34	"	54	6	74	J
35	#	55	7	75	K
36	$	56	8	76	L
37	%	57	9	77	M
38	&	58	:	78	N
39	'	59	;	79	O
40	(	60	<	8Ø	P
41	)	61	=	81	Q
42	*	62	>	82	R
43	+	63	?	83	S
44	,	64	@	84	T
45	-	65	A	85	U
46	.	66	B	86	V
47	/	67	C	87	W
48	Ø	68	D	88	X
49	1	69	E	89	Y
5Ø	2	70	F	9Ø	Z
51	3	71	G	91	[

Código	Cáracter
92	/
93	]
94	↑
95	_
96	`
97	a
98	b
99	c
1ØØ	d
1Ø1	e
1Ø2	f
1Ø3	g
1Ø4	h
1Ø5	i
1Ø6	j
1Ø7	k
1Ø8	l
1Ø9	m
11Ø	n
111	o
112	p
113	q
114	r
115	s
116	t
117	u
118	v
119	w
120	x
121	y
122	z
123	{
124	|
125	}
126	~

EJERCICIOS RESUELTOS

X.1. *Escribir un programa mediante el cual entrando una palabra cualquiera la escriba al revés.*

Solución

El programa es:

```
10   INPUT "La palabra es  "; A$
20   FOR n=LEN(A$) TO 1 STEP -1
30   PRINT MID$ (A$,n,1);
40   NEXT n
50   END
```

* * *

X.2. *Escribir un programa para obtener mensajes cifrados utilizando las sentencias ASC y CHR$ sumando 3 a cada código.*

Solución

El programa es:

```
10    INPUT "El mensaje es  ";A$
20  ┌ FOR I=1 TO LEN (A$)
30  │ LET p$=MID$(A$,I,1)
40  │ LET p$=CHR$(ASC(p$)+3)
50  │ PRINT p$;
60  └ NEXT I
70    END
```

Si A$ es el mensaje:

"El jueves a las 6"

imprimirá

Ho#mxhyhv#d#odv#9

El programa para descifrar este mensaje sería el mismo cambiando en la línea 40, +3 por -3.

* * *

X.3. *Escribir un programa que dé los números capicúas menores que 100000000 que sean cuadrados perfectos (números políndromos).*

Solución

El programa es:

```
 10   ┌ FOR a=10 TO 10000
 20   │ LET b=a*a
 30   │ LET c$=STR$(b)
 40   │   ┌FOR i=1 TO LEN(c$)
 50   │   │LET n$(i)=MID$(c$,i,1)
 60   │   └NEXT i
 70   │ LET d$=""
 80   │   ┌FOR k=LEN(c$) TO 1 STEP -1
 90   │   │LET d$=d$+n$(k)
100   │   └NEXT k
110   │ IF VAL(c$) < > VAL(d$) THEN GO TO 130
120   │ PRINT "El capicua  ";b;"  es el cua
      │ drado de   ";a
125   │ PRINT
130   └ NEXT a
140     END
```

Los resultados son:

$$121 = 11^2$$
$$484 = 22^2$$
$$676 = 26^2$$
$$10201 = 101^2$$
$$12321 = 111^2$$
$$14641 = 121^2$$
$$40804 = 202^2$$
$$44944 = 212^2$$
$$69696 = 264^2$$
$$94249 = 307^2$$
$$698896 = 836^2$$
$$1002001 = 1001^2$$
$$1234321 = 1111^2$$
$$4008004 = 2002^2$$
$$5221225 = 2285^2$$
$$6948496 = 2636^2$$

* * *

X.4. *Escribir un programa que cuente el número de palabras y el número de letras de un texto de menos de 256 caracteres.*

Solución

El programa es:

```
 5   REM CONTADOR DE LETRAS
 8   PRINT "Con este programa se cuenta el número de
     palabras de una frase y las letras de cada pala
     bra": PRINT
 9   PRINT "Escribe una frase entre comillas"
1Ø   INPUT a$:CLS
2Ø   PRINT a$:PRINT
```

```
 3Ø   DIM c(2Ø)
 4Ø   LET j=Ø
 5Ø  ┌FOR i=1 TO LEN(A$)
 6Ø  │LET x=ASC(MID $(a$,i,1))
 7Ø  │IF ((x < 65 OR x > 90) AND (x < 97 OR x > 122)) AND
     │j < > 0 THEN c(j)=c(j)+1: LET j=Ø: GO TO 1ØØ
 8Ø  │IF ((x < 65 OR x > 90) AND (X < 97 OR x >122)) AND
     │j = Ø THEN GO TO 1ØØ
 9Ø  │LET j=j+1
1ØØ  └NEXT i
11Ø   IF j<>Ø THEN LET c(j)=c(j)+1
12Ø   PRINT "En esta frase hay:": PRINT
125   LET m=Ø
13Ø  ┌FOR n=1 TO 2Ø
14Ø  │IF c(n)< >Ø THEN PRINT c(n);"  palabras de  ";n;
     │"  letras"
145  │LET m=m+c(n)
15Ø  └NEXT n
16Ø   PRINT : PRINT "y en total  ";m;"  palabras"
17Ø   END
```

* * *

EJERCICIOS PROPUESTOS

323. ¿Qué imprimirá el siguiente programa?

```
10  LET A$="El precio de una entrada es de
    350 ptas"
20  PRINT MID$(A$,4,6)
30  PRINT
40  PRINT TAB(10); MID$(A$,18,7)
50  PRINT
60  PRINT TAB(20); RIGT$(A$,8)
70  END
```

*324. ¿Qué imprimirá la ejecución del siguiente programa?

```
10  LET A$="TEN ESPERANZA":LET B$="POR TELE"
20  LET C$="A  21  76 ANTES DE":LET D$="A LAS 18"
30  PRINT LEFT$(A$,2);MID$(A$,4,6);MID$(B$,2,1);
40  PRINT MID$(B$,4,1);MID$(B$,6,2);
50  PRINT MID$(C$,2,4);LEFT$(D$,6);
60  PRINT RIGHT$(D$,1)
70  END
```

325. Escribir un programa que sirva para imprimir una cadena a$ de menos de 33 caracteres, subrayada.

326. Escribir un programa para imprimir de un texto de 100 palabras las que empiezan por a y acaban en b estando las palabras del texto en líneas DATA.

*327. Escribir un programa para que en medio de la pantalla aparezca letra por letra una palabra de a

lo sumo 20 letras.

328. Escribir un programa para que aparezca en pantalla letra por letra la frase:

NO FUME POR FAVOR

y que una vez escrita desaparezca y vuelva a repetirse.

329. Escribir un programa para que cuente cuántas vocales minúsculas tiene un texto dado.

330. Escribir un programa para que cambie todas las vocales minúsculas por una vocal introducida de antemano.

*331. Escribir un programa que imprima todas las palabras que empiezan por P de un texto con un número de caracteres inferior a 256. (En el texto hay que colocar un espacio antes de los signos , ; . ? ! ...)

332. Escribir un programa para que imprima mensajes cifrados con claves distintas del problema resuelto número 2 de este tema.

333. Escribir un programa que intercale una cadena B$ entre el de lugar I e I+1 de una cadena A$.

*334. Escribir un programa que cuente cuántas veces se repite la cadena B$ dentro de la cadena A$.

*335. Escribir un programa en el que dados el nombre (no compuesto) y los dos apellidos de una persona en este orden, escriba el primer apellido luego el segundo y a continuación sólo la inicial del nombre seguido de un punto.

336. En una línea DATA están almacenados los nombres de N personas en la forma siguiente: 1er apellido 2º apellido Nombre.

Escribir un programa en el que introduciendo el primer apellido imprima el nombre completo (1er apellido 2º apellido Nombre) de todas las personas con este primer apellido.

*337. Escribir un programa que compare una quiniela con el resultado de la jornada y que imprima el número de aciertos de la misma.

338. Una peña quinielística acostumbra a hacer 5Ø quinielas cada jornada y las repite en la jornada siguiente. Escribir un programa en el que entrando la quiniela con el resultado de la jornada, imprima el número de quinielas de 14 aciertos, el de 13, el de 12 y el de 11, de entre las 50.
(Nota: Las quinielas fijas de la peña, guardadas en un DATA)

339. Escribir un programa que imprima la primera letra de cada palabra que hay en una frase de menos

de 256 caracteres.

*340. Escribir un programa para que calcule la raíz cuadrada de un número e imprima el resultado con 37 cifras.

ANEXO

1. Sentencia INKEY$ (tecla)

La sentencia INPUT sirve para entrar datos y detiene la ejecución del programa esperando que se introduzcan los datos pedidos.

Existe otra sentencia para la entrada de caracteres, es la sentencia INKEY$. Esta sentencia es parecida a la sentencia INPUT pero no detiene la ejecución del programa.

Cuando al ejecutarse un programa, éste llega a la línea

```
n LET A$ = INKEY$
```

el ordenador repasa el teclado y si en este instante se tiene pulsada una tecla, asigna a A$ el carácter que corresponde a la tecla pulsada. Si en este instante no se pulsa ninguna, entonces A$ es la cadena vacía.

Es muy frecuente utilizar INKEY$ en la forma siguiente:

```
n LET A$ = INKEY$ : IF A$ = " " THEN GO TO n
```

o bien:

```
n  IF INKEY$ = " " THEN GO TO n
m  LET A$ = INKEY$
```

Estas líneas forman un bucle que sólo se termina pulsando una tecla a fin de asignar un caracter a la variable A$.

Obsérvese que las líneas del tipo anterior son como INPUT pero no hace falta pulsar ENTER o RETURN. Esto es particularmente útil en programas de juegos.

2. Sentencias POKE y PEEK

Cuando asignamos valores a variables numéricas

A,B,C,..., o alfanuméricas A$,B$,C$,..., el ordenador almacena éstos en posiciones de su memoria RAM. Estas posiciones de memoria están numeradas y el ordenador almacena también en ellas las líneas de programa.

Algunos segmentos de memoria están reservados para contener variables especiales. Por ejemplo, información sobre la pantalla, sobre los sonidos que pueden producir las teclas, etc...

Existe en BASIC una sentencia que puede modificar estas posiciones de memoria especiales.

a) **Sentencia POKE (hurgar)**

Se utiliza de la siguiente forma:

POKE M,N

donde M es un número que indica una posición de memoria y N es el número que queremos poner en la posición M. O sea, POKE M,N almacena el valor N en la posición M ($0 \leq N \leq 255$).

Naturalmente para usar la sentencia POKE es necesario conocer las posiciones especiales de memoria del ordenador (mapa de memoria) y el significado que tienen los números almacenados en éstas. Este conocimiento junto con la aplicación adecuada de la sentencia POKE proporciona un gran dominio del ordenador. Hay que decir también que usar POKE erróneamente tiene un gran poder destructor en los programas.

b) **Sentencia PEEK (mirar)**

La sentencia PEEK actúa de forma inversa a POKE.

Se utiliza de la forma siguiente:

PEEK (M)

y da el número N almacenado en la posición de memoria M.

Si hacemos primero POKE 23609,100 y luego pulsamos:

PRINT PEEK (23609), imprimirá 100

En resumen PEEK (M) sirve para mirar el contenido de la posición de memoria M.

3. Grabación y carga de programas

Tal como se ha indicado, al desconectar el ordenador queda borrada la memoria RAM. También es conveniente, en general, borrar un programa antes de almacenar otro en el ordenador. Es necesario pues guardar los programas en memorias no volátiles, exteriores al ordenador (memorias de masa). Los dispositivos más simples y baratos son las cintas de cassetes aunque los mejores son los discos.

Se utilizan en general dos comandos para grabar programas y cargarlos respectivamente que son

SAVE (o CSAVE) (guardar) y LOAD (o CLOAD) (cargar).

Si tenemos un programa en la memoria RAM del ordenador, se puede grabar en cassete haciendo:

```
SAVE "...  ..."
          ↑
          Nombre del programa.
```

Para volver a cargar el programa desde la cinta al ordenador se utiliza:

```
LOAD "... ..."
          ↑
          Nombre del programa.
```

Estos comandos pueden ser algo distintos en cada ordenador, y es necesario, una vez más, consultar el manual para utilizarlos correctamente.

4. Ficheros

Las líneas DATA utilizadas hasta ahora sirven para almacenar datos conocidos antes de la ejecución de un programa pero no para grabar datos obtenidos como resultado de este programa. Otro inconveniente de la sentencia DATA es que para modificar datos hay que cambiar líneas de programa lo cual resulta engorroso.

Para un tratamiento más preciso de datos se utilizan los ficheros,

a) **Ficheros secuenciales**

Son los únicos que se pueden utilizar en cintas cassete, en ellos los datos están almacenados unos detrás de otro en el orden que se van escribiendo.

Las sentencias para utilizar ficheros son en general:

OPEN, CLOSE, PRINT# e INPUT#

Aunque la utilización de estas sentencias varía según el tipo de ordenador daremos unas indicaciones generales. (Debe consultarse el manual de cada ordenador).

OPEN (abrir)

Para utilizar un fichero es necesario primero abrirlo y para esto sirve la **sentencia OPEN.**

OPEN indica al ordenador que se abre un fichero y debe escribirse a continuación:

El nombre del fichero

El número del fichero

Un indicador para saber si vamos a escribir datos en este fichero o por el contrario vamos a leer datos ya escritos en él.

La forma en que se dan estas indicaciones **depende** del ordenador pero las líneas que utilizan OPEN son de la forma:

```
n OPEN "TELE.";4;for output as
```

o bien

```
n OPEN "TELE." for  output as#4
```

e indicarán al ordenador que se abre el fichero TELE., cuyo número es el 4 para escribir o guardar datos en él.

En algún ordenador en lugar de escribir for output as, bastará escribir un código (por ejemplo 1) que es el de **escribir.**

```
n OPEN "TELE.";4;for input as
```

indicará que abrimos el fichero TELE. de número 4 para leer datos de él. (For input as puede estar sustituído por un código de **lectura**).

PRINT#

PRINT# se utiliza para escribir (guardar) datos en un fichero. Así:

```
n PRINT# 4, A,B,C$
```

escribirá (grabará) en el fichero 4 los valores de

A,B y C$. O sea, PRINT# actúa como PRINT (sin #) pero en el fichero en lugar de la pantalla.

Es importante recordar que para utilizar PRINT#4 es necesario que en una línea anterior se haya abierto el fichero 4 con la indicación de que se abre para escribir en él.

INPUT#

INPUT# se utiliza para leer datos de un fichero asignando estos datos a unas variables.

Así,

n INPUT# 4, A,B,C$

asignará a las variables A,B,C$ los valores que estaban guardados en el fichero 4. Es obvio que el tipo de variables (numéricas o alfanuméricas) guardadas deben corresponderse con las de la línea INPUT#.

Para utilizar INPUT# es necesario que en una línea anterior se haya abierto el fichero 4, con la indicación de que es para leer datos de él.

CLOSE (cerrar)

La sentencia CLOSE sirve para cerrar el fichero después de haber escrito o leído datos.

Basta escribir

n CLOSE# k

para que el ordenador cierre el fichero k.

Ejemplo

El programa siguiente serviría para crear un fichero de 100 nombres y sus correspondientes teléfonos.

```
10  OPEN "TELE." for output as#5
20 ┌FOR I=1 TO 100
30 │INPUT "Nombre",N$
40 │INPUT "Telefono",T
50 │PRINT#5,N$,T
60 └NEXT I
70  CLOSE#5
80  END
```

Para leer los datos del fichero anterior utilizaremos el programa

```
 5  DIM N$(100) : DIM T(100)
10  OPEN "TELE." for INPUT A$#5
20 ┌FOR I=1 TO 100
30 │INPUT#5;N$(I),T(I)
40 └NEXT I
50  CLOSE#5
60  END
```

Después de ejecutarlo, las variables N$(I) contendrían los nombres y las T(I) los correspondientes teléfonos.

Insistimos, una vez más, en que hay pequeñas diferencias en cada modelo de ordenador respecto al uso de OPEN, CLOSE, PRINT# e INPUT#.

b) Ficheros de Acceso directo

Para poder modificar los datos de un fichero secuencial es necesario volcar éstos en la memoria RAM y crear otro fichero con los datos modificados.

Estos problemas resultan mucho más fáciles en en los ficheros de acceso directo. Estos se realizan en unidades de discos (generalmente discos flexibles o floppy disks). El estudio de las unidades de discos y de los ficheros de acceso directo escapa de las posibilidades de este texto.



SOLUCION DE LOS EJERCICIOS

SOLUCION DE LOS EJERCICIOS

Tema I

1. No, en base 6 el equivalente al número decimal 6 es el 10.

2. a) $10_{(2}$; b) $101_{(2}$; c) $111_{(2}$; d) $1100_{(2}$

3. 157.

4. $200_{(3}$.

5. $101000100_{(2}$

6. a) $223232_{(8}$; b) $10010011010011010_{(2}$

7. $346_{(7} = 10110101_{(2}$; $250_{(7} = 10000101_{(2}$

8. $475_{(8} = 10331_{(4}$; $274_{(8} = 2330_{(4}$

9. $1\beta\beta166_{(12}$

10. a) $55041_{(6}$; b) $4481_{(12}$

11. 4024734

12. $10001_{(2} = 17$

13.

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	10
2	2	3	4	5	6	10	11
3	3	4	5	6	10	11	12
4	4	5	6	10	11	12	13
5	5	6	10	11	12	13	14
6	6	10	11	12	13	14	15

×	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	11	13	15
3	0	3	6	12	15	21	24
4	0	4	11	15	22	26	33
5	0	5	13	21	26	34	42
6	0	6	15	24	33	42	51

14. $10110_{(2}$

15. $10010000100_{(2}$

16. $111100_{(2}$

17. $5566_{(8}$

18. 4743

19. Cociente: 124 , resto: 236

20. 4

21. 8

22. 6

23. $184826_{(9}$

24. $x = 7$

25. $x = 4$

26. $x = 4$

27. $x = 3$

28. $0,221_{(6}$

Tema II

29. ..

30. ..

31. a) Contiene programas y funciones grabadas por el fabricante.
    b) No

32. No

33. La respuesta correcta es la b)

34. a) 1Ø682.916
    b) 98367573

35. a) Ø.3619375
    b) 1,164435 E-9

36. a) Ø.11Ø8246
    b) 4,5491Ø89

37. a) 3.8641593 E+1Ø
    b) 1,3142168 E+8

38. a) 41.24
    b) 125.42917

39. ...

40. a) 1,2 seg. ; b) 5.2359878 rad/seg
    c) Ø.8333333 Hertz ; d) 16.493362 cm/seg
    e) 86.359Ø4 cm/seg^2

41. a) Ø.85458725 rad ; b) 1.2210069 rad ;
    c) 1,18885Ø3 rad

42. a) Ø.43388374 ; b) Ø.72654253

43. a) a = 1.389151 rad ; b) a = 1.39Ø1686 rad ;
    c) a = 1.Ø636325 rad

44. ...

45. a) 22.459158 ; b) 23.140693

46. a) 1.5Ø4Ø774 ; b) 6.29319

47. a) Ø.3Ø1Ø3 ; b) 2.158Ø154

48. -Ø.ØØ68578

49. -2.389Ø128

50. Mensaje de error por entrar en la expresión una potencia de base negativa.

51. a) 15ØØ625 ; b) 1555.8Ø96

52. Mensaje de error.

53. Mensaje de error

54. a) 41Ø ; b) 156.25

55. a) 835 ; b) 34

56. PRINT INT(RND * 100)+1

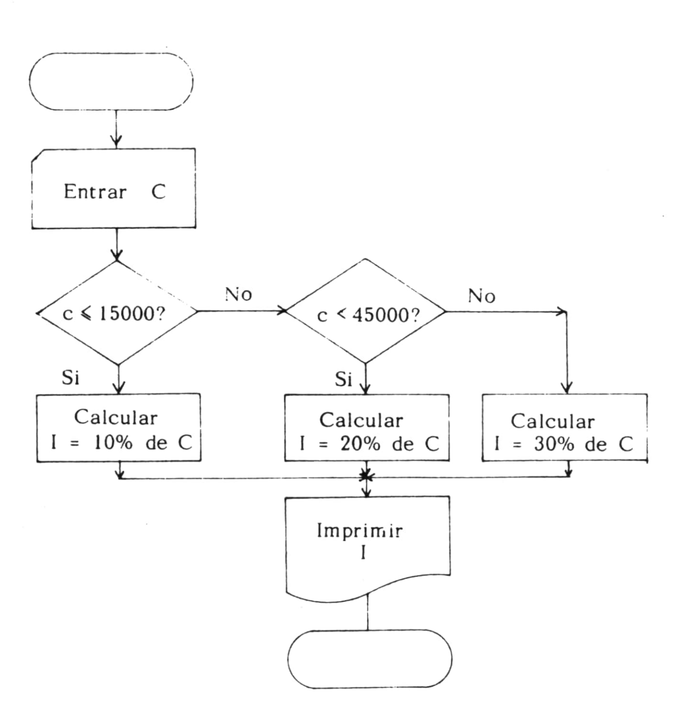
57. 1000

58. a) Un número aleatorio entre 0 y 100($0 \leqslant x < 100$)
    b) Un número entero aleatorio entre 1 y 10 ($1 \leqslant x \leqslant 10$)

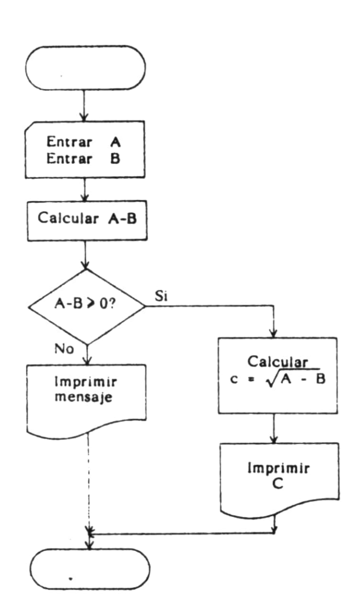
Tema III

59. La respuesta correcta es la c), pues las comparaciones lógicas sólo admiten como resultado la afirmación o la negación.

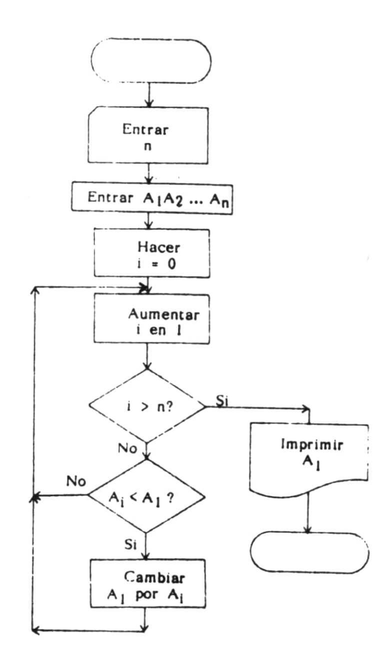
63.



67.


Entrar A
Entrar B
Calcular A-B
A-B ≥ 0?
Si
No
Imprimir mensaje
Calcular c = √A - B
Imprimir C

70.


Entrar n
Entrar A1A2 ... An
Hacer i = 0
Aumentar i en 1
i > n?
Si
No
Imprimir A1
No
Ai < A1 ?
Si
Cambiar A1 por Ai

Nótese que se comparan todos con A_1 y cada vez que se encuentra uno menor que A_1 se coloca en su lugar.

El A_1 resultante es el menor.

77. El área del cuadrado de lado 5 es igual a 25.

80. La linea 4Ø es errónea, pues no pueden multiplicarse las variables alfanuméricas.

82.
```
1Ø PRINT A$
2Ø PRINT
3Ø PRINT B$
4Ø PRINT
5Ø PRINT C$
```

91. a) El programa es:
```
1Ø LET A = 3 : LET B = 4.5
2Ø PRINT((531.2*A)↑2 * ABS(TAN(2*π*B/3))↑(1/5)
3Ø END
```
b) Basta escribir nuevamente la línea 1Ø de la siguiente forma:
```
1Ø LET A = 7.3 : LET B = 59.8
```

93. Nos daría un mensaje de error de variable no definida o bien

Ø
Ø

según el modelo de ordenador utilizado.

96. 2   75   46

97. A*2 = 5Ø
    A+A = 5Ø

    pues al repetir la línea 3Ø sin ninguna sentencia queda eliminada la anterior.

100. No hay errores. Su ejecución da como resultado

```
5Ø
4
4ØØ
```

107. La 3Ø es incorrecta puesto que la asignación está puesta al revés, debe ser LET X = A+B

111.

```
1Ø LET M = A   (sirve para guardar la varia-
2Ø LET A = B    ble A)
3Ø LET B = M
```

Tema IV

117. Esperará que el usuario le proporcione un número y una cadena de caracteres.

120. Sí, pues en la línea 3Ø , A↑2 sólo lo efectuará si A ⩾ 0, luego debe ser

```
30 PRINT A * A
```

123.

```
10 REM Datos personales
20 INPUT "Nombre"; A$
30 INPUT "Primer apellido" ; B$
40 INPUT "Segundo apellido"; C$
50 INPUT "EDAD" ; D
60 INPUT "Número carnet" ; E
70 PRINT "Nombre: " ; A$
80 PRINT
90 PRINT
100 PRINT "Primer apellido: " ; B$
110 PRINT
120 PRINT
130 PRINT "Segundo apellido: " ; C$
140 PRINT
150 PRINT
160 PRINT "Edad: " ; D
170 PRINT
180 PRINT
190 PRINT "Número carnet: "; E
200 END
```

125. a) Hay un error de sintaxis en la línea 3Ø. Debe ser

```
30 LET Y = A/B
```

b) Hay un error de sintaxis en la línea 3Ø. Debe ser

```
30 LET Y = A + B/A * C
```

c) No hay errores.

137.

```
10 REM División entera
20 INPUT "Dividendo =" ; a
25 INPUT "DIVISOR =" ; b
30 LET Z = a/b
40 PRINT "El cociente es  " ; INT(Z)
50 PRINT "El resto es  " ; a - b * INT(Z)
60 END
```

140.

```
10 INPUT "Primer término =" ; a
20 INPUT "Diferencia =" ; d
30 INPUT "Número de términos =" ; n
40 LET u = a + (n-1) * d
50 PRINT "La suma es S =  "; (a+u) * n/2
60 END
```

148.

```
10 INPUT "Longitud en cm. =" ; l
20 LET T = 2 * π * (l/981) ↑ 0.5
30 PRINT "Para l = " ; l ; "  cm , el perío-
   do es T = " ; T
40 END
```

152.

```
10 INPUT Vo , a , t
20 REM Componentes de la velocidad
30 PRINT Vo * COS(a), Vo * SIN(a)-9.81 * t
40 REM Coordenadas del proyectil
50 PRINT Vo * t * COS(a), Vo * t SIN(a)-9.81
   * t ↑ 2/2
60 END
```

Tema V

155. No, pues siempre se ejecutará la línea 15Ø

158. No lo podemos asegurar, puesto que en alguna línea anterior podría tener una orden de salto a la 11Ø.

161. 1Ø IF A/2 = INT(A/2) AND A > 4Ø THEN PRINT A

168.
```
1Ø INPUT A, B, C
2Ø IF B< A THEN LET A = B
3Ø IF C< A THEN LET A = C
4Ø PRINT "El menor es ";A
5Ø END
```

169.
```
1Ø INPUT a,b,c,d,e,f,g,h,i,j
2Ø PRINT "A continuación hay que pulsar la
   clase de mercancia que se vende"
3Ø INPUT M$
4Ø IF M$ = "A1" THEN LET a = a - 1
5Ø IF M$ = "A2" THEN LET b = b - 1
6Ø IF M$ = "A3" THEN LET c = c - 1
7Ø IF M$ = "A4" THEN LET d = d - 1
8Ø IF M$ = "A5" THEN LET e = e - 1
9Ø IF M$ = "A6" THEN LET f = f - 1
1ØØ IF M$ = "A7" THEN LET g = g - 1
11Ø IF M$ = "A8" THEN LET h = h - 1
12Ø IF M$ = "A9" THEN LET i = i - 1
13Ø IF M$ = "A1Ø" THEN LET j = j - 1
```

```
140 INPUT "Hay mas ventas a contabilizar?";V$
150 IF V$ = "Si" THEN GO TO 30
160 PRINT "Quedan en total:"
170 PRINT a;"  de A1" , b;"  de A2"
180 PRINT c;"  de A3" , d;"  de A4"
190 PRINT e;"  de A5" , f;"  de A6"
200 PRINT g;"  de A7" , h;"  de A8"
210 PRINT i;"  de A9" , j;"  de A10"
220 END
```

175.

```
10 PRINT "Grados"; TAB(8);"Seno";TAB(20);"Co-
   seno"
20 LET X = 0
30 PRINT
40 PRINT TAB(3);X;TAB(8);SIN(X*π/180);
   TAB(20);COS(X*π/180)
50 LET X = X + 1
60 IF  X <= 90 THEN GO TO 30
70 END
```

178.

```
10 LET n = 1
20 PRINT "a";n;"  =  "; (1 + 1/n) ↑ n
30 LET n = n + 1
40 GO TO 20
50 END
```

180. En la división entera

$$\begin{array}{l|l} D & d \\ \hline r & c \end{array}$$

se verifica: r = D - d.c, siendo c = INT(D/d) y en nuestro caso d = 4

```
10 INPUT n
20 LET r = n - 4*INT(n/4)
30 IF r = 0 THEN PRINT "i elevado a  ";n;
   "  =  ";1
40 IF r = 1 THEN PRINT "i elevado a  ";n;
   "  =  ";i
```

```
50 IF r = 2 THEN PRINT "i elevado a ";n;
   " = ";-1
60 IF r = 3 THEN PRINT "i elevado a ";n;
   " = ";-i
70 GO TO 10
80 END
```

184.

```
10 LET F = 0 : LET M = 0
20 LET n = INT(RND * 100) + 1
30 IF n> = 41 AND n< = 60 THEN F = F + 1
40 IF M = 1000 THEN GO TO 100
50 LET M = M + 1
60 GO TO 20
100 PRINT "La frecuencia relativa es ";F/1000
110 PRINT
120 PRINT "La diferencia entre la frecuencia
   relativa y la probabilidad es ";F/1000-0.2
130 END
```

189.

```
10 INPUT n
20 LET F = 1
30 IF n = 1 THEN GO TO 70
40 LET F = F * n
50 LET n = n - 1
60 GO TO 30
70 PRINT "Factorial de ";n;" es ";F
80 END
```

Este programa se realizará de otra manera en el próximo tema.

191.

```
10 INPUT "Numero a descomponer";n
20 LET a = n : LET c = 1
30 LET d = n/2
40 IF d = INT(d) THEN PRINT 2 : LET c =
   = c * 2 : LET n = d : GO TO 30
50 LET k = 3
60 LET d = n/k
```

```
70 IF d = INT(d) THEN LET n = d : PRINT
   k : LET c = c*k : GO TO 60
80 LET k = k+2
90 IF k > a/2  AND c = 1 THEN PRINT a;
   "  es primo" : GO TO 120
100 IF c = a THEN GO TO 120
110 GO TO 60
120 END
```

Tema VI

197. No, pues la línea 2Ø junto con la 4Ø harán que n aumente de 4 en 4.

200. Para números menores o iguales que 6 imprime 10 veces este número y luego él mismo aumentado en 4.

Para números mayores que 6 escribe el número y luego mensaje de error puesto que encontrará un NEXT sin FOR.

204. Escribirá Ø indefinidamente pues debido a la línea 2Ø y a la 4Ø el valor de n siempre en Ø.

211.

```
1Ø ┌ FOR X = 3 TO 2Ø
2Ø │ PRINT X ↑ 2 + 1
3Ø └ NEXT X
4Ø   PRINT "Fin"
5Ø   END
```

216.

```
1Ø   INPUT n
2Ø   LET S = 1
3Ø ┌ FOR i = 1 TO n
4Ø │ LET S = S + (½) ↑ i
5Ø └ NEXT i
6Ø   PRINT "S =  ";S
7Ø   END
```

218.

```
10   INPUT "El primer término es";al
20   INPUT "La razón es";r
30   INPUT "El número de términos es";N
40   LET L = al / (1-r)
50   PRINT "El límite de la suma es  ";L
60   LET S = al
70 ┌ FOR i = 1 TO N-1
80 │ LET S = S + al * r ↑ i
90 └ NEXT i
100  PRINT "La suma de los  ";N;"  primeros
     términos es  ";S
110  PRINT "La diferencia entre el limite y
     la suma de los  ";N;"  primeros términos
     es  ";L-S
120  END
```

219.

```
10   INPUT N
20   LET A = 0
30 ┌ FOR i = 1 TO N
40 │ LET X = RND * 10
50 │ LET Y = RND * 10
60 │ IF SQR ((X-5) * (X-5) + (Y-5) * (Y-5))< 5
   │  THEN LET A = A + 1
70 └ NEXT i
80   PRINT "La frecuencia relativa es  ";A/N
90   PRINT "Un valor aproximado de pi es  ";
     4 * A/N
100  END
```

224.

```
10 ┌FOR C = 3 TO 40
20 │ ┌FOR A = 2 TO 39
30 │ │ ┌FOR B = 1 TO A - 1
40 │ │ │ IF A * A + B * B = C * C THEN PRINT
   │ │ │    "A="; A, "B="; B, "C="; C
50 │ │ └NEXT B
60 │ └NEXT A
70 └ NEXT C
80   END
```

227.

```
10   INPUT "Número de puntos y radio"; n,R
20  ┌FOR a = Ø TO 2*π  STEP(2*π)/n
30  │PRINT "Las coordenadas son:"
40  │PRINT "X = "; R*COS(a),
    │" Y= "; R*SIN(a)
50  └NEXT a
60   END
```

229.

```
10  ┌FOR a = 1 TO 9
20  │ ┌FOR b = Ø TO 9
30  │ │LET x = a*1ØØ + b*1Ø + a
40  │ │LET y = 2*a*a + b*b
50  │ │IF x = y THEN PRINT x
60  │ └NEXT b
70  └NEXT a
80   END
```

Ejecutando el programa se observa que no hay ninguno.

235.

```
10   INPUT "Número"; n
20   PRINT "Los divisores de  ";n;"  son:"
30   PRINT
40  ┌FOR i = INT(n/2) TO 1 STEP -1
50  │IF (n/i) < > INT(n/i) THEN GO TO 7Ø
60  │PRINT i
70  └NEXT i
80   PRINT n
90   END
```

Tema VII

241. Sí puesto que en la línea 3Ø se asigna a una variable alfanumérica un dato numérico.

243. En la línea 1Ø debe ser A,B
En la línea 2Ø debe ser LET X = A + B
En la línea 4Ø debe ser 23,1Ø

248. Imprimirá: 2,4,-3,2,-4,-3,7.2

252. Da la temperatura de cada día de la semana y la temperatura media, que en este caso es 19,85 grados.

254.

```
1Ø ┌FOR n = 1 TO 2Ø
2Ø │READ A,B,C,D
3Ø │PRINT A,B,C,D
4Ø │RESTORE
5Ø └NEXT n
6Ø  DATA 1,2,3,4,5,6,7
7Ø  END
```

258.

```
1Ø READ L,Ma,Mi,J,V,S,D
2Ø LET S = L+Ma+Mi+J+Vi+S+D
3Ø PRINT "Gasto total =  ";S
4Ø PRINT "Gasto medio =  ";S/7
5Ø DATA ...(gastos de cada día de la semana)
6Ø END
```

260.

```
10   LET A = Ø : LET B = Ø
20 ┌ FOR i = 1 TO N
30 │ READ X
40 │ IF X > 1.7Ø LET A = A+1
50 │ IF X < 1.6Ø LET B = B+1
60 └ NEXT i
70   PRINT "Número de personas de altura
     mayor que 1.7Ø =  ";A
80   PRINT "Número de personas de altura
     menor que 1.6Ø =  ";B
90   DATA ...(altura de las N personas)
1ØØ  END
```

264.

```
1Ø   READ p$
2Ø ┌ FOR i = TO 9
3Ø │ READ c$
4Ø │ IF c$ < p$ THEN LET p$ = c$
5Ø └ NEXT i
6Ø   PRINT "EL primero es:  ";p$
7Ø   DATA "PEDRO" , "JUAN" , "JORDI" ,
     "XAVIER" , "DAMIAN" , "DAVID" ,
     "PEPE" , "LUIS" , "ORIOL" , "MANEL"
8Ø   END
```

265.

```
1Ø ┌ FOR N = 1 TO 1Ø
2Ø │ ┌ FOR I = 1 TO 1Ø
3Ø │ │ READ A
4Ø │ │ IF A = 1 THEN PRINT TAB(I+1Ø); "*";
5Ø │ └ NEXT I
6Ø │ PRINT
7Ø └ NEXT N
8Ø   DATA 1,Ø,1,1,1,1,1,1,1,1
9Ø   DATA 1,Ø,1,1,1,1,Ø,Ø,Ø,1
1ØØ  DATA 1,Ø,Ø,Ø,Ø,1,Ø,1,1,1
11Ø  DATA 1,1,Ø,1,Ø,1,Ø,1,1,1
12Ø  DATA 1,1,Ø,1,Ø,1,Ø,1,1,1
13Ø  DATA 1,1,Ø,1,Ø,Ø,Ø,Ø,Ø,1
14Ø  DATA 1,1,Ø,1,1,1,Ø,1,Ø,1
```

```
150  DATA 1,1,0,1,1,1,0,1,0,1
160  DATA 1,0,0,1,1,0,0,1,0,1
170  DATA 1,1,1,1,1,1,1,1,0,1
180  END
```

269.

```
10   PRINT : PRINT
20   READ N$,C$
30   IF N$ = "F" THEN PRINT "No tengo memorizados
     más países": GO TO 150
50   PRINT "La capital de  "; N$; " es?"
60   FOR I=1 TO 3
70   INPUT X$
80   IF X$   C$ THEN PRINT "No es correcto"
     IF I   3  THEN PRINT "Prueba otra vez"
90   IF X$ = C$ THEN PRINT "Muy bien": GO TO 10
100  NEXT I
110  CLS
120  PRINT "La capital de  ";N$;" es ";C$
130  GO TO 10
140  DATA "E.E.U.U." , "Washington" , "Canadá" ,
     "Otawa" , "México", "México D.F." , "Guatemala",
     "Guatemala" , "El Salvador" , "San Salvador" ,
     "Honduras"  ,  "Tegucigalpa"  ,  "Nicaragua"  ,
     "Managua"
150  DATA "Costa Rica" , "San José" , "Panamá"
     "Panamá" , "Cuba" , "La Habana" , "Rep.
     Dominicana" , "Santo Domingo" , "Colombia"
     "Bogotá" , "Venezuela" , "Caracas" , "Ecuador",
     "Quito"
160  DATA "Perú" , "Lima" , "Brasil" , "Brasilia" ,
     "Paraguay" , "Asunción" , "Uruguay" , "Montevi
     deo" , "Argentina" , "Buenos Aires" , "Chile" ,
     "Santiago"
170  DATA "Bolivia" , "La Paz" , "Jamaica" , Kings
     ton" , "Haití" , "Puerto Príncipe" , "Guayana
     Francesa" , "Cayena" , "F" , "F"
180  END
```

271.

```
10   PRINT "Las materias son:"
20   PRINT "1 FISICA" : PRINT "2 QUIMICA" :
     PRINT "3 FISIOLOGIA Y MEDICINA" : PRINT
     "4 LITERATURA" : PRINT "5 LA PAZ" : PRINT
     "6 ECONOMIA"
30   PRINT
40   INPUT "Número de la materia";A
45   CLS
50   PRINT "Los premios Nobel en la década de
     los 70 en esta materia son:"
60   FOR i = 1 TO 112
70   READ n , E$
80   IFn = A THEN PRINT E$
90   NEXT i
100  DATA 1, "Louis Neil (Francia)" ,2, "Luis F.
     Leloir (Argentina)", 3, "Bernard Katz (Alema-
     nia), Julius Axelrod (EE.UU), Ulf von Euler
     (Suecia)", 4, "Alexander Solzhenitsyn (URSS)" ,
     5, "Norman Borlaug (EE.UU.)"
110  DATA 1, "Dennis Gabor (Gran Bretaña)", 2,
     "Gerhard Herzberg (Canadá)", 3, "Earl Suther
     land (EE.UU.)" , 4, "Pablo Neruda (Chile)",
     5, "Willy Brandt (Alemania)", 6, "Simon Kuznets
     (EE.UU.)"
120  DATA 1, "John Bardeen (EE.UU.), Leon L.
     Cooper (EE.UU.), John R. Schrieffer (EE.UU.)",
     2, "Christian B. Anfinsen (EE.UU.), Stanford
     Moore (EE.UU.) , William Stein (EE.UU.)"
130  DATA 3, "Gerard Edelman (EE.UU.) , Rodney
     R. Porter (Gran Bretaña)", 4, "Heinrich Böll
     (Alemania)", 6, "John R. Hicks (Gran Bretaña),
     Kenneth J. Arrow (EE.UU.)"
140  DATA 1, "Brian Josephson (Gran Bretaña)
     , Leo Esaki (Japón) , Ivar Giaever (EE.UU.)"
     2, "Ernst O. Fischer (Alemania), Geoffrey
     Wilkinson (Gran Bretaña)"
150  DATA 3, "Nikolaas Tinbergen (Holanda), Karl
     von Frisch (Austria) , Konrad Lorenz (Austria)",
     4, "Patrick White (Australia)", 5, "Henry A.
     Kissinge (EE.UU.), Le Duc Tho (Vietnam N.)",
     6, "Wassily Leontief (EE.UU.)"
```

```
16Ø DATA 1, "Martín Ryle (Gran Bretaña), Antony
    Herwish (Gran Bretaña)", 2, "Paul J. Flory
    (EE.UU.)", 3, "George Emil Palade (EE.UU.),
    Albert Clauder (Bélgica), Christian de Duve
    (Gran Bretaña)"
17Ø DATA 4, "H. Martinson (Suecia), Eyvin Johnson
    (Suecia)", 5, "Eisaku Sato (Japón) , Séan Mc.Bri
    de (Irlanda)", 6, "Karl G. Myrdal (Suecia),
    Friedrich August von Hayek (Gran Bretaña)"
18Ø DATA 1, "Aage N. Bohr (Dinamarca), Ben
    R. Motlelson (Dinamarca), James Rainwate
    (EE.UU.)", 2, "John W. Cornforth (Gran Bretaña)
    Wladimir Prelog (Suiza)"
19Ø DATA 3, "Renato Dulbecco (EE.UU.) , David
    Baltimore (EE.UU.), Howard Temin (EE.UU.)",
    4, "Eugenio Montale (Italia)", 5, "Andrei Sajarof
    (URSS)", 6, "Lenidid V. Kantorovich (URSS), Tja
    lling C. Koopmans (Holanda)".
2ØØ DATA 1, "Philip W.Anderson (EE.UU.) , John
    van Vlek (EE.UU.) , 2, "Ilya Prigonine (Bélgica)
    3, "Rosalyn Yalow (EE.UU.) , Roger Guillemin
    (EE.UU.) , Andrew Schally (EE.UU.)"
21Ø DATA 4, "Vicente Aleixandre (España)" ,5,
    "Amnistía Internacional" ,6, "Berthil Ohlin
    (Suecia) , James Meade (Gran Bretaña)"
22Ø DATA 1, "Piotr L. Kapitsa (URSS), Arno A.
    Penzias (EE.UU.) , Robert W. Wilson ( EE.UU.)",
    2, "Peter Mitchell (Gran Bretaña)", 3, "Werner
    Arber (Suiza) , Daniel Nathaus (EE.UU.) ,
    Hamilton O. Smith (EE.UU.)"
23Ø DATA 4, "Isaac B. Singer (EE.UU.)" ,5, "Mena-
    hem Begin (Israel) , Anwar-el Sadat (Egipto)",
    6, "Herbert A. Simon (EE.UU.)"
24Ø DATA 1, "Sheldon L. Glashow (EE.UU.), Steven
    Weinberg (EE.UU.) , Abdus Salam (Pakistán)",
    2, "Herbert Ch. Brown (EE.UU.) , Georg Wittig
    (Alemania federal)", 3, "Altan M. Cormacck
    (EE.UU.), Godfrey N. Heunsfield (Gran Bretaña)"
25Ø DATA 4, "Odysseus Elytis (Grecia)",5, "Teresa
    de Calculta", 6, "Theodore W. Schultz (EE.UU.),
    Arthur Lewis (Gran Bretaña)"
```

```
26Ø  DATA 1, "James W. Cronin (EE.UU.) , Val
     L. Fitch (EE.UU.)", 2, "Paul Berg (EE.UU.) ,
     Walter Gilbert (EE.UU.) , Frederic Sanger (Gran
     Bretaña)" ,3, "Baruj Benacerraf (EE.UU.) ,
     George Snell (EE.UU.) , Jean Dausset (Francia)"
27Ø  DATA 4, "Caeslaw Milosz (Polonia)", 5, "Adolfo
     Pérez Esquivel (Argentina)", 6, "Lawrence R.
     Klein (EE.UU.)"
28Ø  END
```

Tema VIII

277. No porque en la subrutina encontraría un RETURN sin GOSUB.

281. La superficie del cuadrado de lado 2 es 4
La superficie del cuadrado de lado 5 es 25
La superficie del cuadrado de lado 8 es 64
La superficie del cuadrado de lado 9 es 81

282. El programa es correcto. No importa donde se definan las funciones.

284.

```
 10  PRINT "S1=1"
 20  LET S=1
 30 ┌FOR n=1 TO 14
 40 │GOSUB 100
 50 │LET m=1/k
 60 │LET S=S+m
 70 │PRINT "S";n+1;"=";S
 80 └NEXT n
 90  GO TO 150
100  LET k=1
110 ┌FOR x=1 TO n
120 │LET k=k*x
130 └NEXT x
140  RETURN
150  PRINT "La diferencia entre S15 y e es  ";S-EXP(1)
160  END
```

293.
```
10  DEF FN f(a,b,c)=2*a+3*b+4*c+2*a*b+3*b*c
20 ┌FOR a=0 TO 7
30 │  ┌FOR b=3 TO 14
40 │  │  ┌FOR c=5 TO 12
50 │  │  │PRINT "a=";a;TAB(7);"b=";b;TAB(14);"c=";c;
   │  │  │TAB(21);"z="; FNf (a,b,c,) + FN f(a,b,c) ↑ 2 +
   │  │  │FN f(a,b,c) ↑ 3
60 │  │  └NEXT c
70 │  └NEXT b
80 └NEXT a
90  END
```

295.
```
100  REM CALCULO DE f COMPUESTA CON g
110  LET z=FN f(x)
120  LET y=FN g(z)
130  RETURN
```

298.
```
10  INPUT "Entrar a y b"; a,b
20  INPUT "Entrar c y d"; c,d
30  PRINT TAB(14);"MENU"
35  PRINT: PRINT
40  PRINT "1 calcular la suma"
50  PRINT "2 calcular el producto"
60  PRINT "3 calcular el cociente"
70  INPUT "Opción deseada";m
80  ON m GOSUB 100,200,300
85  IF m < 1  OR  m>3  THEN GO TO 70
90  PRINT "El resultado es";R;"+(";I;")i"
95  GO TO 340
100 REM SUMA
110 LET R=a+c : LET I=b+d
120 RETURN
200 REM PRODUCTO
210 LET R=a*c-b*d : LET I=a*d+b*c
220 RETURN
300 REM COCIENTE
310 LET R=(a*c+b*d)/(c*c+d*d)
320 LET I=(b*c-a*d)/(c*c+d*d)
330 RETURN
340 END
```

TEMA IX

302. Imprimirá todos los números de la línea DATA y también la media aritmética (M.A.) de ellos que es 4,1266666

306.
```
10  DIM A(3) : DIM B(3)
20 ┌FOR I=1 TO 3
30 │PRINT "Componente ";I;" de A?":INPUT A(I)
40 │PRINT "Componente ";I;" de B?":INPUT B(I)
50 └NEXT I
60  PRINT "El producto escalar es A.B=";A(1)*B(1)+
    A(2)*B(2)+A(3)*B(3)
70  PRINT "Las componentes del producto vectorial
    son (";A(2)*B(3)-A(3)*B(2);",";A(3)*B(1)-A(1)*B(3);","
    ;A(1)*B(2)-A(2)*B(1);")"
80  END
```

311.
```
10  DIM A(8,5)
20 ┌FOR I=1 TO 8
30 │    ┌FOR J=1 TO 5
40 │    │INPUT A(I,J)
50 │    │PRINT TAB(6*J-5);A(I,J);
60 │    └NEXT J
70 │ GOSUB 200
80 └NEXT I
90  PRINT : PRINT "La tabla con las filas ordenadas
    es:"
```

```
100  FOR I=1 TO 8
110      FOR J=1 TO 5
120      PRINT TAB(6*J-5);A(I,J);
130      NEXT J
140  NEXT I
150  GO TO 270
200  REM SUBRUTINA DE ORDENACION
210  FOR M=2 TO 5
220      FOR N=1 TO M-1
230      IF A(I,M) < A(I,N) THEN LET X=A(I,N):
         LET A(I,N)=A(I,M):LET A(I,M)=X
240      NEXT N
250  NEXT M
260  RETURN
270  END
```

314.
```
10  DIM A(40):DIM B(40):DIM M(40)
20  REM Entrar las notas
30  FOR I=1 TO 40
40  INPUT "Primera y segunda nota";A(I),B(I)
50  LET M(I)=(A(I)+B(I))/2
60  PRINT "La nota media del alumno  ";I;" es ";M(I)
70  NEXT I
80  END
```

316.
```
10  DIM N(30):DIM B$(6)
20  FOR I=1 TO 30
30  INPUT N(I)
40  IF N(I)< 2.5  THEN LET B$(I)=B$(1)+"*"
50  IF N(I)>=2.5  AND N(I)<5 THEN LET B$(2)=B$(2)+
    "*"
60  IF N(I)>=5 AND N(I)< 6 THEN LET B$(3)=B$(3)+"*"
70  IF N(I)>=6 AND N(I) < 7 THEN LET B$(4)=B$(4)+"*"
80  IF N(I)>=7 AND N(I)<=8 THEN LET B$(5)=B$(5)+"*"
90  IF N(I) > 8 THEN LET B$(6)=B$(6)+"*"
100 NEXT I
110 FOR k=1 TO 6
120 READ C$
130 PRINT C$;TAB(4);B$(k)
140 PRINT : PRINT
150 NEXT k
```

```
160  DATA "MD","I","S","B","N","E"
170  END
```

Para los ordenadores que funcionan según la nota 2 de la pregunta 4 de este tema, el programa es:

```
10   DIM N(30) : DIM a$(6,30) : DIM S(6)
20  ┌FOR i=1 TO 30
30  │INPUT N(i)
40  │IF N(i)<2.5 THEN LET S(1)=S(1)+1 :
    │LET a$(1,S(1))="*"
50  │IF N(i)>=2.5 AND N(i)<5 THEN LET S(2)=S(2)+1 :
    │LET a$(2,S(2))="*"
60  │IF N(i)>=5 AND N(i)< 6 THEN LET S(3)=S(3)+1 :
    │LET a$(3,S(3))="*"
70  │IF N(i)>=6 AND N(i) < 7 THEN LET S(4)=S(4)+1 :
    │LET a$(4,S(4))="*"
80  │IF N(i)>=7 AND N(i)<=8 THEN LET S(5)=S(5)+1 :
    │LET a$(5,S(5))="*"
90  │IF N(i) > 8 THEN LET S(6)=S(6)+1 :
    │LET A$(6,S(6))="*"
100 └NEXT i
110 ┌FOR k=i TO 6
120 │READ c$
130 │PRINT c$ ; TAB(4) ; a$(k)
140 │PRINT : PRINT
150 └NEXT k
160  DATA "MD","I","S","B","N","E"
170  END
```

322.

```
10   REM PERMUTACIONES CON 5 LETRAS DISTIN-
     TAS
15   DIM A$(5)
20   PRINT "Que letras quieres permutar?":PRINT
30   INPUT A$(1),A$(2),A$(3),A$(4),A$(5)
35   PRINT A$(1),A$(2),A$(3),A$(4),A$(5):PRINT:PRINT
```

```
40 ┌FOR a=1 TO 5
50 │   ┌FOR b=1 TO 5
60 │   │IF b=a THEN GO TO 150
70 │   │   ┌FOR c=1 TO 5
80 │   │   │IF c=a OR c=b THEN GO TO 140
90 │   │   │   ┌FOR d=1 TO 5
100│   │   │   │IF d=a OR d=b OR d=c THEN GO
   │   │   │   │TO 130
110│   │   │   │LET e=15-(a+b+c+d)
120│   │   │   │PRINT A$(a);A$(b);A$(c);A$(d);A$(e)
130│   │   │   └NEXT d
140│   │   └NEXT c
150│   └NEXT b
160└NEXT a
170 END
```

Tema X

324. TE ESPERO EL 21 A LAS 8

327.

```
1Ø  INPUT "La palabra es";A$
15  IF LEN(A$)>2Ø THEN GO TO 1Ø
2Ø  LET L=LEN(A$)
3Ø  FOR I=1 TO L
4Ø  PRINT TAB(15-L/2);MID$(A$,I,1);
5Ø  PAUSE(5)
6Ø  NEXT I
7Ø  END
```

En algún ordenador hay que sustituir PAUSE por WAIT. Se podría poner también:

```
FOR n=1 TO 3ØØ : NEXT n
```

331.

```
1Ø   INPUT "Texto entre comillas"; F$
15   IF LEFT$(F$,1) ="p" OR MID$(F$,1,1)="P" THEN
     LET i=Ø: GOSUB 1ØØ
2Ø   FOR i=1 TO LEN(F$)-1
3Ø   IF MID$(F$,i,1)=" "AND(MID$(F$,i+1,1)="p" OR
     MID$(F$,i+1,1)="P") THEN GOSUB 1ØØ
4Ø   NEXT i
1ØØ  PRINT
11Ø  PRINT MID$(F$,i+1,1);
12Ø  IF MID$(F$,i+1,1)=" " THEN RETURN
```

```
130  LET i=i+1
140  IF i<LEN(F$) THEN GO TO 110
150  END
```

334.

```
10  INPUT A$
20  INPUT B$
30  LET N=LEN(B$): LET T=0
40 ┌FOR I=1 TO LEN(A$)-N+1
50 │LET C$=MID$(A$,I,N)
60 │IF C$=B$ THEN LET T=T+1
70 └NEXT I
80  PRINT B$; "  se repite  ";T;"  veces en  ";A$
90  END
```

335.

```
10  INPUT "Nombre y apellidos";N$
20  LET I$=LEFT$(N$,1)
30 ┌FOR I=2 TO LEN(N$)
40 │IF MID$(N$,I,1)=" "  THEN GO TO 60
50 └NEXT I
60  LET M$=RIGHT$(N$,LEN(N$)-I)
70  PRINT M$+" "+I$+"."
```

337.

```
10  PRINT "Introducir los 14 resultados en fila"
15  LET n=0
20  INPUT A$:PRINT
30  PRINT "Mi quiniela es:"
40  INPUT B$: PRINT
50 ┌FOR i=1 TO 14
60 │IF MID$(A$,i,1)=MID$(B$,i,1) THEN n=n+1
70 └NEXT i
80  PRINT : PRINT "Hay  ";n;" aciertos"
90  END
```

340.
```
 10  REM RAIZ CUADRADA CON 37 CIFRAS
 20  INPUT "Número  ";a
 30  LET b=INT(SQR(a))
 40  LET a$=STR$(b)
 50  LET k=LEN(a$)
 60  PRINT "La raíz cuadrada de   ";a;"  es",
 70  PRINT : PRINT
 80  PRINT b;".";
 90  LET c=a-b*b
100  LET d=100*c
110  LET e=INT((d/10)/(2*b))
120  LET f=(2*b*10+e)*e
130  IF f>d THEN LET e=e-1 : GO TO 120
140  PRIN e; : LET k=k+1
150  IF k=37 THEN GO TO 180
160  LET g=d-f : LET c=g : LET b=10*b+e
170  GO TO 100
180  END
```

INDICE

PPU
Promociones Publicaciones Universitarias